REDUCED-SET MODELS FOR IMPROVING THE TRAINING AND EXECUTION SPEED OF KERNEL METHODS

A Thesis
Presented to
The Academic Faculty

by

Hassan A. Kingravi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2014

REDUCED-SET MODELS FOR IMPROVING THE TRAINING AND EXECUTION SPEED OF KERNEL METHODS

Approved by:

Professor Mark A. Davenport,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Vijay K. Madisetti
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Patricio A. Vela, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Girish Chowdhary
Mechanical and Aerospace engineering
Oklahoma State University

Professor Erik Verriest
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 10th December 2013

To my parents.

ACKNOWLEDGEMENTS

First and foremost, I'd like to thank my advisor Patricio Vela for his support and guidance over the last three years. I learned a great deal from him, not just about the research that we worked on, but also about the steps one needs to take in order to formulate and solve problems in general. In this sense, he taught me by leading by example, and for that, I'm grateful.

I'd like to thank Vijay Madisetti for agreeing to be on my thesis committee on such short notice. I'm grateful to Mark Davenport for taking time out from his insanely busy schedule to be on my committee, for teaching me about some fundamental topics in his advanced DSP class, and for his penchant for amusing wordplay and puns, which was unfortunately lost on most of the grad students in his class. Erik Verriest has been a source of inspiration ever since I came to the ECE department, due to his wonderful classes, his deep insight into mathematical structure, and his willingness to answer any questions I had on the more esoteric aspects of estimation and control. I'm also indebted to Girish Chowdhary, as he is one of the people I most closely collaborate with, and because without him, the second half of this thesis would not exist in its current form. Alexander Gray provided valuable input into some of the ideas in this thesis, and wrote to me when I was in the hospital, which I appreciated. The late Scott Wills was instrumental in convincing me to switch to ECE, and was a source of hope during some very difficult times.

A lot of the work over the past few years has been made possible by outside collaborators. My friend Emre Celebi helped me get into research many years ago, and has (with his better half, Farida Hamza) provided moral and material support ever since I've known him. Our collaborators at LIDS at MIT have made some of

the material in the second half of the thesis possible, particularly Rob Grande and Jonathan How.

My thanks also goes out to my labmates, Gbolabo Ogunmakin, Miguel Serrano, Guangcong Zhang, and Leo Keselman. Gbolabo was a good person to discuss music with, Guangcong has been a valuable resource to bounce research ideas off of, Leo has been a valuable resource to bounce a literal ball off of, and Miguel has great hair. Our office would have been quite dull without you guys. I thank Ivan Kolesov for being a valued semi-member of the lab, for helping me approach the tricky world of weight lifting, and for helping me break my hand in the pursuit of the same (that's how you know it's working). I benefited greatly from my friendship with Nishant Mehta, as he provided valuable insights into the cutting edge of machine learning, and who was a fun person to hang out with outside of work. I'd also like to acknowledge Smriti Chopra, for being a good friend over the past few years, and for her insistence on inviting me to events that we both know I will never attend.

I'd like to thank my two closest friends at Tech in the last few years: Damien Rontani was a pleasure to hang out and discuss all kinds of topics with, ranging from research to politics. Ali Ahmad and I suffered through many courses together, engaged in countless brotherly arguments, and helped encourage one another at the numerous low points in our academic careers. I'd also like to acknowledge the Pakistani community at large, particularly Murtaza Askari, Ali Hashmi, Farooq Akram, Bashir Akbar, and Adeel Yousuf, who helped me adjust to life at Tech.

More than anyone else, I'd like to thank my family. My parents have supported me unconditionally throughout my academic career. My brother Ali has helped me out on numerous occasions, and my sister Hina has taken care of me ever since she came to the US. Finally, I'd like to thank my lady, Mehtab Wasi: her belief in me, and her love and support ever since I've known her is something I'll always be grateful for.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis aims to contribute to the area of kernel methods, which are a class of machine learning methods known for their wide applicability and state-of-the-art performance, but which suffer from high training and evaluation complexity.

The work in this thesis utilizes the notion of *reduced-set models* to alleviate the training and testing complexities of these methods in a unified manner. In the first part of the thesis, we use recent results in kernel smoothing and integral-operator learning to design a generic strategy to speed up various kernel methods. In Chapter 3, we present a method to speed up kernel PCA (KPCA), which is one of the fundamental kernel methods for manifold learning, by using reduced-set density estimates (RSDE) of the data. The proposed method induces an integral operator that is an approximation of the ideal integral operator associated to KPCA. It is shown that the error between the ideal and approximate integral operators is related to the error between the ideal and approximate kernel density estimates of the data. In Chapter 4, we derive similar approximation algorithms for Gaussian process regression, diffusion maps, and kernel embeddings of conditional distributions.

In the second part of the thesis, we use reduced-set models for kernel methods to tackle online learning in model-reference adaptive control (MRAC). In Chapter 5, we relate the properties of the feature spaces induced by Mercer kernels to make a connection between persistency-of-excitation and the budgeted placement of kernels to minimize tracking and modeling error. In Chapter 6, we use a Gaussian process (GP) formulation of the modeling error to accommodate a larger class of errors, and design a reduced-set algorithm to learn a GP model of the modeling error. Proofs of stability for all the algorithms are presented, and simulation results on a challenging

control problem validate the methods.

CHAPTER I

INTRODUCTION

Machine learning is a scientific field which focuses on the creation and study of systems that learn automatically from what can be loosely labeled as 'observations'. Fundamentally, machine learning is the algorithmic base that enables machines (or agents, as they are sometimes known) to operate *autonomously*, while adapting to unforeseen changes in their environment [7]. What differentiates machine learning from older theories for autonomous systems, such as control theory, is that the aforementioned adaptation is created on the basis of a *model* or a *learned representation* of the world, as observed through the system's sensors. Figure 1 shows an example of an autonomous system, and the place of machine learning in the system.

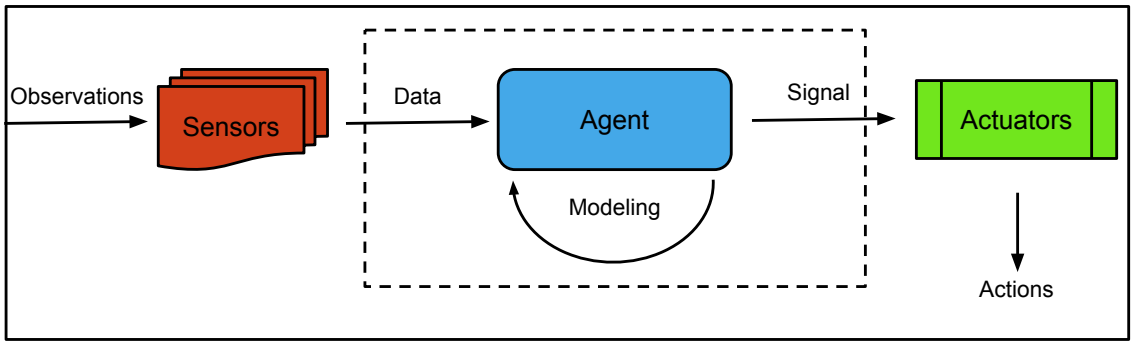


Figure 1: A pictorial representation of a generic autonomous system. Machine learning constitutes the middle portion of the diagram.

There are different subfields in machine learning, each of which is designed to tackle a different type of problem. The focus of this thesis is on statistical machine learning, which uses the mathematics relevant to statistics to learn from data. The tools developed in the field have become increasingly important for extrapolation from data in the 21st century; examples of data of interest to practitioners in the field

include images from websites, time-series data generated from the weather, speech corpora, analytics from social media, etc. The roots of machine learning can be traced back to the 1950s and 1960s, which saw the introduction of algorithms for detecting linear relations among data [81, 92]. The development of nonlinear algorithms accelerated greatly in the 1980s, which saw the introduction of the backpropagation algorithm for neural networks [83], and an induction algorithm for decision trees [76]. Decision trees and neural networks proved extremely effective for a wide variety of applications, partially because these algorithms were directly inspired by the way humans themselves learn. Decision trees can be thought of as an abstract representation of choices and their possible outcomes, a method of decision-making that is very commonly used by humans in practice, and the roots of neural networks are inspired by neuroscientific principles. There are two issues with these classes of algorithms however: 1) they are based on heuristics, and at the time of their introduction, their statistical behavior was not well understood, and 2) in most cases, they are difficult to train, and/or assume a predefined structure imposed by the designer of the system. The mid 1990s saw the introduction of support-vector machines (SVMs) [11], which led to a new revolution in machine learning via the introduction of nonlinear feature mappings induced by kernel functions, which effectively allowed a straightforward conversion of linear methods into nonlinear ones, by a process known as *kernelization* [23]. The success of SVMs led to the kernelization of methods such as ridge regression, PCA, and adaptive filtering, among many others [92]. At the same time, the statistical performance of kernel methods was studied using tools from empirical processes, tools which were pioneered by Vapnik and Chervonenkis in the 1960s, and which laid a firm theoretical foundation for these methods [106]. In addition to their solid theoretical footing, the training of kernel methods boils down to well-understood optimization techniques such as convex optimization and eigendecomposition, which eschews the tendency of older optimization algorithms to get stuck in local optima

[11].

Kernel methods thus have many attractive features: 1) kernels can be designed for specific classes of problems, and in many cases, a single type of kernel function suffices, 2) their training is well understood, and 3) they let the 'data speak for themselves', by computing hypotheses that rely on kernel expansions on the data itself. Unfortunately, this nonparametric nature is the biggest liability of these methods in practice, since a dataset with n points results in a target function that requires $\mathcal{O}(n)$ function evaluations for every new test point [92]. In the data deluge associated with modern computing, n can easily exceed millions of training points, causing these methods to become impractical at test time. Even worse, their training complexity can be up to $\mathcal{O}(n^3)$, which precludes training on datasets larger than a few thousand points on most personal computers. To alleviate this heavy computational burden, a great deal of effort has been devoted to making the training and testing phases of kernel methods efficient separately [27, 89]; however, these approaches do not try to treat these phases as a whole.

The approach outlined in the first part of the thesis attempts to unify training and testing speedups via the simplest solution possible: *by discarding a large portion of the data before the actual training procedure even takes place.* It is readily apparent that this is an extremely aggressive strategy to pursue, and the goal of the thesis is to outline an intelligent data reduction strategy that is guaranteed to not lose too much accuracy. To do so, the general method takes a (deterministic) 'sketch' of the data before discarding it. This sketch is then used to compute reduced-set approximations to the hypotheses that would have been computed using the full training set. We demonstrate the effectiveness of these reduced-set approximations on a wide variety of machine learning problems, such as classification, regression, clustering and kernel embeddings, by proving rigorous bounds on loss in approximation quality, and by performing experiments on a wide variety of datasets.

In the second part of the thesis, we show how related reduced-set approximations can be used to learn time-series data. In the field of adaptive control, controllers must adapt to a controlled system with significant modeling error and/or disturbances. One common method for this adaptation is model-reference adaptive control (MRAC), where this error, also known as the uncertainty, can be learned using radial basis function networks. While effective in practice, MRAC usually assumes knowledge of the domain of the uncertainty, and/or focuses on minimizing tracking error between the actual and the desired model. The work in this thesis uses tools from reproducing-kernel Hilbert space theory and Gaussian processes to create new classes of controllers, which do not require a priori knowledge of the uncertainty, and whose goal is to actually learn a nonparametric model of the uncertainty, so as to combat it more effectively. It is shown that the new methods effectively give these controllers *memory*, and that this memory allows them to anticipate future errors, and thus minimize control effort on the actuators. Lyapunov-like analysis is used to guarantee stability of the controllers, and simulation results on a challenging nonlinear control problem are presented, showing improvement in tracking error and a long-term learning effect.

The remainder of this thesis is organized into the following chapters. See Figure 2 for chapter dependencies.

Chapter 3: Reduced-Set Kernel PCA: Basic Theory This chapter presents a method for speeding up the training and execution speed of kernel learning algorithms relying on spectral decomposition. First, the connection of kernel smoothing to the spectral decomposition of integral operators is exploited, within the context of KPCA, to define reduced-set KPCA, which relies on the computation of a reduced-set density estimate (RSDE) of the dataset, with a much smaller cardinality than the full set. The RSKPCA approach circumvents the computation of the full kernel matrix, so that the eigendecomposition and kernel mapping is performed using the RSDE alone. Results bounding the approximation of the density via the maximum mean discrepancy and

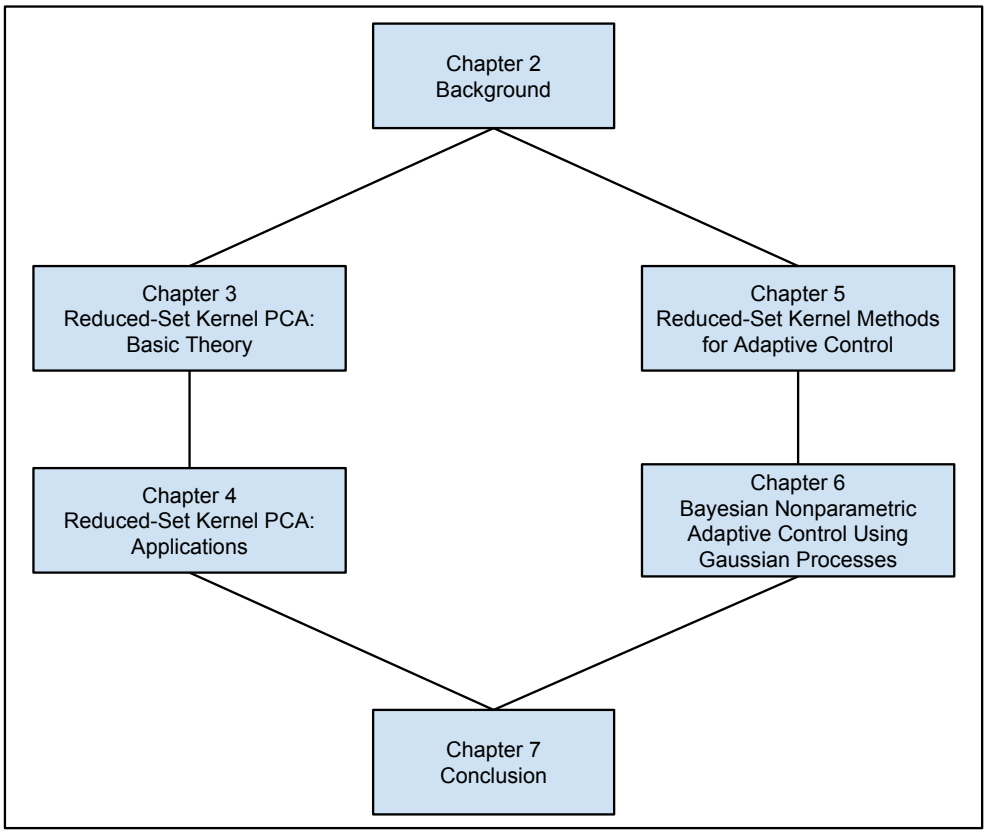


Figure 2: Structure of the thesis.

the difference in Hilbert-Schmidt norm between the operators are presented, providing further theoretical justification for the approach. In the case of radially-symmetric kernels, a simple, single-pass RSDE called the shadow density estimate is presented, which allows for the computation of closed-form bounds on the approximation errors mentioned above. Experimental results on various datasets are given which show the effectiveness of the approach in practice, for speeding up both training and execution speed, as well as yielding good approximation accuracy.

Chapter 4: Reduced-Set Kernel PCA: Applications This chapter uses the theory developed in the previous chapter to compute reduced-set approximations for three other types of kernel methods. First, an algorithm for Gaussian process regression is presented, and theoretical results bounding the deviation of the hypothesis of the reduced-set approximation from the original hypothesis are presented. Second,

we present an approximation for the diffusion map algorithm, and present similar theoretical results. Third, approximation algorithms are given for different applications of kernel embeddings, including the kernel Bayes' rule. Finally, experimental results showing the efficacy of each of these approximation algorithms are given.

Chapter 5: Reduced-Set Kernel Methods for Adaptive Control Classical work in model reference adaptive control for uncertain nonlinear dynamical systems with an RBF neural network adaptive element does not guarantee that the network weights stay bounded in a compact neighborhood of the ideal weights when the system signals are not persistently exciting (PE). Recent work has shown, however, that an adaptive controller using specifically recorded data concurrently with instantaneous data guarantees boundedness without PE signals. However, the work assumes fixed RBF network centers, which requires domain knowledge of the uncertainty. Motivated by reproducing kernel Hilbert space theory, in this chapter, we propose an online algorithm for updating the RBF centers to remove the assumption. In addition to proving boundedness of the resulting neuro-adaptive controller, a connection is made between PE signals and kernel methods. Simulation results show improved performance.

Chapter 6: Bayesian Nonparametric Adaptive Control Using Gaussian Processes This chapter investigates a Gaussian process (GP) based Bayesian MRAC architecture (GP-MRAC), which leverages the power and flexibility of GP Bayesian nonparametric models of uncertainty. GP-MRAC does not require the centers of the RBF to be preallocated, can inherently handle measurement noise, and enables MRAC to handle a broader set of uncertainties, including those that are defined as distributions over functions. We use stochastic stability arguments to show that GP-MRAC guarantees good closed loop performance with no prior domain knowledge of the uncertainty. Online implementable GP inference methods are compared in numerical simulations against RBFN-MRAC with preallocated centers and are shown

to provide better tracking and improved long-term learning.

Chapter 7: Conclusion This chapter concludes the thesis, and offers ideas where to proceed from here.

Table 1: Table of Notation

$\mathbb{R}^d$	Euclidean space of dimension d.
$\mathbb{C}^d$	Complex space of dimension d.
$\mathbb{M}$	Compact metric space.
$\mathscr{C}_0(\Omega)$	Space of continuous functions on the domain Ω.
$\mathscr{C}^k(\Omega)$	Space of kth differentiable continuous functions on the domain Ω.
$\mathscr{C}^\infty(\Omega)$	Space of infinitely differentiable continuous functions on the domain Ω.
$\mathcal{L}^p(\Omega)$	Space of all functions f on domain ω such that $\left(\int_\Omega \lvert f\rvert^p dx\right)^{\frac{1}{p}} < \infty$ with respect to Lebesgue measure.
$\mathcal{L}^p(\Omega, \nu)$	Space of all functions f on domain ω such that $\left(\int_\Omega \lvert f\rvert^p dx\right)^{\frac{1}{p}} < \infty$ with respect to measure ν.
$\mathcal{H}$	Reproducing-kernel Hilbert space.
$\mathcal{H}^*$	Dual space of $\mathcal{H}$.
$\widehat{f}$	Fourier transform of function $f \in \mathcal{L}^1(\Omega)$.
$\mathrm{MMD}(X, Y)$	Maximum mean discrepancy between random variables X and Y.
$\lVert \cdot \rVert_{\mathrm{op}}$	Operator norm.
$\lVert \cdot \rVert_{\mathrm{HS}}$	Hilbert-Schmidt norm.

CHAPTER II

BACKGROUND

This section reviews some of the basic concepts used in this thesis. We mainly employ tools from real and functional analysis as applied to kernel methods, as well as basic results from probability and control theory.

2.1 Measure Theory

This section references some of the concepts used in the thesis from analysis; a good source for this material is [30]. Measure spaces are denoted by the triple $(\Omega, \mathcal{A}, \upsilon)$, where Ω is the domain, $\mathcal{A}$ is a σ-algebra, and υ is an abstract measure. Recall that a measure is called σ-*finite* if $\Omega = \bigcup_{i=1}^{\infty} S_i$, where $S_i \in \mathcal{A}$ and $\upsilon(S_i) < \infty$ for all i. If f is a υ-measurable function on Ω, a new measure ν can be defined using the integral

$$\nu = \int_{\Omega} f d\upsilon. \tag{2.1.1}$$

This measure will be referred to using the notation

$$d\nu = f d\upsilon. \tag{2.1.2}$$

We will also need to utilize some ideas from the theory of Radon measures. In particular, given a compact metric space $\mathbb{M}$, we denote the set of continuous functions on $\mathbb{M}$ by $\mathscr{C}_0(\mathbb{M})$, and its dual by $\mathscr{C}_0(\mathbb{M})^*$. The weak topology on $\mathscr{C}_0(\mathbb{M})$ is the unique weakest topology on $\mathscr{C}_0(\mathbb{M})$ which makes all elements in $\mathscr{C}_0(\mathbb{M})^*$ continuous, while the weak* topology on $\mathscr{C}_0(\mathbb{M})^*$ is defined in a similar manner with respect to $\mathscr{C}_0(\mathbb{M})^{**}$. These facts allow use to state the following definition.

Definition 2.1.1. *Define the bounded linear functional on* $\mathscr{C}_0(\mathbb{M})$ *with respect to*

measure υ *as*

$$\nu(u) := \int_{\mathbb{M}} u(x) d\upsilon, \tag{2.1.3}$$

where $u \in \mathscr{C}_0(\mathbb{M})$. *Then a sequence* $(\nu_n)_{n\in\mathbb{N}}$ *is said to exhibit* **weak convergence** *to* ν, *denoted* $\nu_n(w) \rightharpoonup \nu(w)$, *if* $|\nu_n(w) - \nu(w)| \to 0 \ \forall w \in \mathscr{C}_0(\mathbb{M})$.

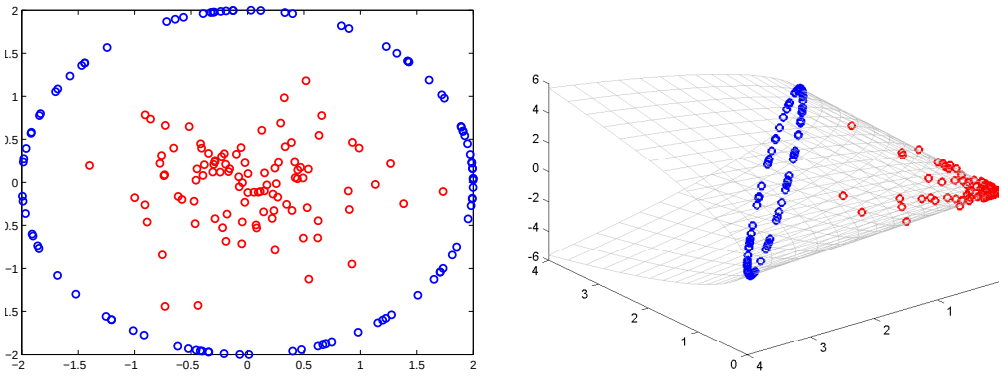
Note that closed, bounded subsets of $\mathbb{R}^d$ are examples of compact metric spaces under the Euclidean metric $d(x, y) = \|x - y\|$.

2.2 Kernel Methods

Kernel methods are a class of machine learning algorithms which use nonlinear feature mappings in conjunction with linear learning algorithms to discover linear relations between observations in feature space. Because of the modular nature of kernels, the net effect is an algorithm which can be nonlinear in the input space. In order to gain an understanding of how an appropriate feature map can help for a learning problem, consider the two sample sets $\mathcal{X}_1 = \{x_i\}_{i=1}^{n_1}$ and $\mathcal{X}_2 = \{x'_j\}_{j=1}^{n_2}$, where $x_i, x'_j \in \mathbb{R}^2$, and the sets are shown in Figure 3a. Consider the problem of learning a decision boundary from the data in the input space which assigns $\mathcal{X}_1$ to class 1 and $\mathcal{X}_2$ to class 2. This problem is an example of a supervised-learning problem. It is impossible to find a linear separator in the input space to achieve 100% training accuracy. However, if the data are mapped to $\mathbb{R}^3$ using the feature map

$$(x, y) \mapsto \left(x^2, y^2, \sqrt{2}xy\right), \tag{2.2.1}$$

finding a linear separator becomes quite easy, as can be seen from Figure 3a. Therefore, in order to solve this classification problem using a two-step approach, one can map the data to the feature space using (2.2.1), and then use any linear classification algorithm, such as a linear perceptron, or a linear SVM, to get the decision boundary in the feature space [7]. The main problem with this approach is that it's not very



(a) Data in input space (b) Data in feature space

Figure 3: A (canonical) example of a classification problem where an appropriate feature map allows a linear separator to achieve 100% training accuracy. The surface shown in the feature space is how the geometry of the input space is warped after projection.

flexible, since the feature mapping needs to be designed by hand. Fortunately, the theory of kernels allows for an elegant solution to this issue.

2.2.1 Reproducing Kernel Hilbert Spaces

The feature spaces employed in kernel methods are classes of function spaces known as reproducing-kernel Hilbert spaces (RKHSs). We outline the technical details concerning these spaces in some detail in this section, so as to facilitate our analysis in later chapters. The majority of the material in this section is based on [111].

Definition 2.2.1. *Let $\mathcal{H}$ be a real Hilbert space of functions $f : \Omega \to \mathbb{R}$. A function $k : \Omega \times \Omega \to \mathbb{R}$ is called a* reproducing kernel *for $\mathcal{H}$ if*

1. $K_y := k(\cdot, y) \in \mathcal{H}, \ \forall y \in \Omega.$
2. $f(y) = \langle f, K_y \rangle_{\mathcal{H}}, \ \forall f \in \mathcal{H}, \ and \ \forall y \in \Omega.$

Note that this definition implies that the kernel associated to a particular $\mathcal{H}$ is unique. Recall that for each Hilbert space $\mathcal{H}$, there is an associated space of continuous functionals known as the *dual space* $\mathcal{H}^*$. Then the following theorem can be proven:

Theorem 2.2.1. *Let $\mathcal{H}$ be a Hilbert space of functions $f : \Omega \times \Omega \to \mathbb{R}$. Then the following are equivalent:*

1. *$\mathcal{H}$ has a reproducing kernel.*

2. *All point evaluation functionals $\delta_y \in \mathcal{H}^*$ are continuous, for all $y \in \Omega$.*

A Hilbert space with associated kernel k is called a reproducing-kernel Hilbert space. In the above theorem, we have $\delta_y := \langle \cdot, K_y \rangle_{\mathcal{H}}$. Somewhat confusingly, the quantities K_y are also known as 'point evaluation functionals', even though they are not strictly functionals, but are associated with their corresponding δ_y's by a linear isometry. It can be shown that

$$k(x, y) = \langle K_x, K_y \rangle_{\mathcal{H}} = \langle \delta_x, \delta_y \rangle_{\mathcal{H}^*} \ \forall x, y \in \Omega. \tag{2.2.2}$$

This property implies the existence of a feature map $\psi : \Omega \to \mathcal{H}$, where $\psi(x) \mapsto K_x$. In particular,

$$k(x, y) = \langle \psi(x), \psi(y) \rangle_{\mathcal{H}}, \ \forall x, y \in \Omega. \tag{2.2.3}$$

What makes RKHSs useful for machine learning is that if a function obeys certain properties, it ensures the existence of this feature map.

Definition 2.2.2. *A function $k : \Omega \times \Omega \to \mathbb{R}$ where Ω is some domain is known as a* kernel function *if it is* positive semidefinite*, i.e.*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j) c_i c_j \geq 0, \tag{2.2.4}$$

for all finite sample sets of size n, where $x_i \in \Omega$ and $c_i \in \mathbb{R}$.

Definition 2.2.2 is equivalent to requiring the kernel (Gram) matrix (i.e. the $\mathbb{R}^{n \times n}$ matrix defined as $K_{ij} := k(x_i, x_j)$), to be positive semidefinite for all finite sample sets where $x_i \in \Omega$. This condition is necessary for the statement of the Moore–Aronszajn theorem, one of the foundational results in the theory of RKHSs. We provide a sketch

of the proof of this theorem for completeness, and to outline some of the subtleties that tend to be omitted in the literature.

Theorem 2.2.2. *(**Moore–Aronszajn** [2]) If k is a symmetric, positive-definite kernel on a set Ω, there exists a unique RKHS $\mathcal{H}$ on Ω, for which k is a reproducing kernel.*

Proof. (**Sketch**) By definition, we know that all functions of the form $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$ lie in $\mathcal{H}$. For these types of functions, we have that

$$\|f\|_{\mathcal{H}}^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \langle k(\cdot, x_i), k(\cdot, x_j)\rangle_k \tag{2.2.5}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j). \tag{2.2.6}$$

Consider the vector space

$$H_k(\Omega) := \operatorname{span}\{k(\cdot, x) : x \in \Omega\},$$

equipped with the bilinear form

$$\left\langle \sum_{i=1}^{n} \alpha_i k(\cdot, x_i), \sum_{i=1}^{m} \beta_j k(\cdot, y_j) \right\rangle_k := \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_i \beta_j k(x_i, y_j).$$

It can be shown that $\mathcal{H}_k(\Omega)$ is a pre-Hilbert space, and that $\langle \cdot, \cdot \rangle_k$ is a valid inner product on the space. The completion $\mathcal{H}_k(\Omega)$ of $H_k(\Omega)$ with respect to the $\|\cdot\|_k$ norm is a natural candidate for the RKHS $\mathcal{H}$. However, since the elements of the completion are abstract elements, we need to define functional values for them. We do so by using the linear mapping

$$R : \mathcal{H}_k(\Omega) \to \mathscr{C}_0(\Omega), \text{ where } R(f)(x) := \langle f, k(\cdot, x)\rangle_k.$$

It is simple to show that all functions of the form Rf, where $f \in \mathcal{H}_k(\Omega)$, are continuous. Furthermore, the mapping $R : \mathcal{H}_k(\Omega) \to \mathscr{C}_0(\Omega)$ is injective. The RKHS $\mathcal{H}(\Omega)$ is then defined by

$$\mathcal{H}(\Omega) := R(\mathcal{H}_k(\Omega)),$$

with inner product

$$\langle f, g\rangle_{\mathcal{H}(\Omega)} := \langle R^{-1}f, R^{-1}g\rangle_k, \ \forall f \in \mathcal{H}(\Omega) \text{ and } x \in \Omega.$$

By construction, $\mathcal{H}_k(\Omega) \subset \mathcal{H}(\Omega)$ is dense in $\mathcal{H}(\Omega)$, and

$$\|f\|_k = \|f\|_{\mathcal{H}(\Omega)}, \ \forall f \in \mathcal{H}_k(\Omega).$$

These last two properties can be used to show the uniqueness of the space $\mathcal{H}(\Omega)$ [1] with respect to k. □

Three common examples of kernel functions are linear (2.2.7), polynomial (2.2.8), and Gaussian kernels (2.2.9).

$$k(x, y) := \langle x, y\rangle_{\mathbb{R}^d}, \tag{2.2.7}$$

$$k(x, y) := (\langle x, y\rangle_{\mathbb{R}^d} + b)^p, \ p \in \mathbb{N}, \ b \in \mathbb{R}, \tag{2.2.8}$$

$$k(x, y) := e^{-\frac{\|x-y\|^2}{2\sigma^2}}, \ \sigma \in \mathbb{R}_+. \tag{2.2.9}$$

Kernels can also be defined on other domains, such as strings, histograms, and probability distributions [100]. If the domain Ω is a subset of $\mathbb{R}^d$ for some dimension d, and the kernel is translation invariant, a powerful characterization of the feature space $\mathcal{H}$ can be proven. We use the notation

$$\widehat{f}(x) := \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(\omega) e^{-j\langle x,\omega\rangle} d\omega, \tag{2.2.10}$$

for the Fourier transform of a function $f \in \mathcal{L}^1(\mathbb{R}^d)$.

Theorem 2.2.3. *Suppose that $k \in \mathscr{C}_0(\mathbb{R}^d) \cap \mathcal{L}^1(\mathbb{R}^d)$ is a real-valued positive definite function. Define*

$$\mathcal{H} := \left\{ f \in \mathcal{L}^2(\mathbb{R}^d) \cap \mathscr{C}_0(\mathbb{R}^d) : \widehat{f} \Big/ \sqrt{\widehat{k}} \in \mathcal{L}^2(\mathbb{R}^d) \right\},$$

[1] We drop the Ω in $\mathcal{H}(\Omega)$ for the rest of the thesis.

and equip the space with the bilinear form

$$\langle f, g\rangle_{\mathcal{H}} := \frac{1}{(2\pi)^{\frac{d}{2}}} \left\langle \widehat{f}\Big/\sqrt{\widehat{k}}, \widehat{g}\Big/\sqrt{\widehat{k}} \right\rangle_{\mathcal{L}^2(\mathbb{R}^d)} = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \frac{\widehat{f}(\omega)\overline{\widehat{g}(\omega)}}{\widehat{k}(\omega)} d\omega. \tag{2.2.11}$$

Then $\mathcal{H}$ is an RKHS with inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and reproducing kernel k. In particular, the inner product (2.2.11) coincides with the inner product induced by k.

Note that this theorem imposes a strong smoothness constraint on RKHSs induced by translation invariant kernels on $\mathbb{R}^d$. For certain kernels, such as the Gaussian, it implies that all functions in the associated $\mathcal{H}$ are $\mathscr{C}^\infty$.

2.2.2 Integral Operators

If $\Omega \subset \mathbb{R}^d$, the kernel induces a linear operator $\mathbf{K} : \mathcal{L}^2(\Omega) \to \mathcal{L}^2(\Omega)$, defined by

$$(\mathbf{K}f)(x) := \int_\Omega k(x, y) f(y) dy, \tag{2.2.12}$$

where $f \in \mathcal{L}^2(\Omega)$, the Hilbert space of square-integrable functions on Ω. Although dy typically corresponds to the Lebesgue measure, (2.2.12) can be generalized to the case where dy is replaced with another measure. The operator (2.2.12) is positive, self-adjoint, and compact for continuous, positive-definite kernels [100]. The following theorem establishes another way of generating feature maps.

Theorem 2.2.4. ***(Mercer** [58]) The eigenvalues $\{\lambda_\iota\}_{\iota=1}^\infty$, and eigenfunctions $\{\phi_\iota\}_{\iota=1}^\infty$ of the operator (2.2.12) obey the following conditions:*

- *The eigenvalues λ_ι are strictly positive and absolutely convergent.*
- *The eigenfunctions ϕ_ι are bounded, and form an orthonormal basis for $\mathcal{L}^2(\Omega)$.*
- *For $x, y \in \Omega$,*

$$k(x, y) = \sum_{\iota=1}^\infty \lambda_\iota \phi_\iota(x) \phi_\iota(y) = \langle \psi(x), \psi(y)\rangle_{\mathcal{H}}. \tag{2.2.13}$$

Corollary 2.2.1. *Let the conditions for Mercer's theorem hold. Then a feature map satisfying (2.2.13) can be defined as*

$$\psi(x) := (\sqrt{\lambda_1}\phi_1(x), \sqrt{\lambda_2}\phi_2(x), \dots,). \tag{2.2.14}$$

We now note that under certain restrictions, the space of operators generated from the kernel also live in a Hilbert space.

Definition 2.2.3. *If the kernel $k : \Omega \times \Omega \to \mathbb{R}$ obeys the restriction*

$$\int_\Omega \int_\Omega |k(x,y)|^2 dxdy < \infty, \tag{2.2.15}$$

it is known as a Hilbert-Schmidt kernel.

Definition 2.2.4. *A bounded operator $A : \mathcal{H} \to \mathcal{H}$ is called a* Hilbert-Schmidt operator *if*

$$\sum_{\iota \geq 1} \|Ae_\iota\|_{\mathcal{H}}^2 < \infty,$$

where $\{e_\iota\}_{\iota \geq 1}$ is an orthonormal basis for $\mathcal{H}$.

Integral operators associated to Hilbert-Schmidt kernels are Hilbert-Schmidt. The following theorem will be useful in later chapters to measure approximate operators.

Theorem 2.2.5. *The space of Hilbert-Schmidt operators $\mathbb{H}$ on Ω is a Hilbert space, with Hilbert-Schmidt norm*

$$\|A\|_{\mathrm{HS}}^2 = \mathrm{Tr}\langle A^*, A\rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \|Ae_i\|_{\mathcal{H}}^2, \tag{2.2.16}$$

The $\|\cdot\|_{\mathrm{HS}}$ norm measures the sum of projections onto any basis in $\mathcal{H}$. In the finite dimensional case, it is represented by the Frobenius norm $\|\cdot\|_F$.

Empirical approximations of the eigendecomposition of (2.2.12) and related integral operators are a primary tool in the design of algorithms such as KPCA [91], Laplacian eigenmaps [5], diffusion maps [22], kernel-ridge regression [87] and least-squares SVMs [92], and are a major focus in this thesis.

2.2.3 The Nyström Method

The Nyström method is a technique for discretizing (2.2.12)[4]. In particular, it is used for solving integral equations of the second kind, which are defined as

$$u(x) - \int_\Omega k(x,y)u(y)dy = w(x),\ x \in \Omega,\ u \in \mathscr{C}_0(\Omega), \tag{2.2.17}$$

for the function u. It does so using approximate measures

$$\upsilon_n(u) = \sum_{x\in\Omega_n} u(x)\alpha_n(x), \tag{2.2.18}$$

where $\Omega_n \subset \Omega$ is a set of finite points or *quadrature nodes*, and $\alpha_n(x) \in \mathbb{R}$ are the weights associated to the procedure. Note that υ_n is a measure on Ω_n, and can be written in the same form as (2.1.1), i.e.

$$\upsilon_n(u) = \int_{\Omega_n} u(x)d\upsilon_n(x),\ u \in \mathscr{C}_0(\Omega_n). \tag{2.2.19}$$

Definition 2.2.5. *The* Nyström approximation method for integral equations of the second kind *finds a function $u_n \in \mathscr{C}_0(\Omega_n)$ satisfying*

$$u_n(x) - \int_{y\in\Omega_n} k_n(x,y)u_n(y)dy = w_n(x),\ x \in \Omega_n, \tag{2.2.20}$$

where k_n and w_n are restrictions of k and w respectively.

The approximations (2.2.20) have empirical operators associated to them, in the form

$$(\mathbf{K}_n u_n)(x) := \int_{y\in\Omega_n} k_n(x,y)u_n(y)d\upsilon_n(y),\ u_n \in \mathscr{C}_0(\Omega_n). \tag{2.2.21}$$

Note that these are finite-dimensional operators $\mathbf{K}_n : \mathscr{C}(\Omega_n) \to \mathscr{C}_0(\Omega_n)$.

Definition 2.2.6. *A sequence of integral operators $(\mathbf{K}_n)_{n\in\mathbb{N}}$ is* consistent at $u \in \mathscr{C}(\Omega)$ *if*

$$\|d_n^k(u)\|_n := \sup_{x\in\Omega_n} |d_n^k(u)(x)| \to 0,\ n \in \mathbb{N}, \tag{2.2.22}$$

where

$$d_n^k(u) := \int_{\Omega_n} k_n(x,y)u(y)d\upsilon_n(y) - \int_\Omega k(x,y)u(y)dy,\ x \in \Omega_n,\ n \in \mathbb{N}. \tag{2.2.23}$$

For the Nyström method, we have that

$$d_n^k(u)(x) = \sum_{y \in \Omega_n} \alpha_n(y) k(x,y) u(y) - \int_\Omega k(x,y) u(y) dy$$
$$= (\upsilon_n - \upsilon)\breve{k}(x, \cdot),\ x \in \Omega_n,\ n \in \mathbb{N}, \tag{2.2.24}$$

where $\breve{k}(x,y) := k(x,y)u(y)$. This fact is used to prove the following theorem.

Theorem 2.2.6. *(**Nyström Approximation Theorem** [79]) Let the weak convergence $\upsilon_n \rightharpoonup \upsilon,\ n \in \mathbb{N}$ be satisfied. Then*

$$\|d_n^k(u)\|_n = \lim_{n \to \infty} \sup_{x \in \Omega_n} \|(\upsilon_n - \upsilon)\breve{k}(x, \cdot)\| \to 0, \tag{2.2.25}$$

and the sequence of integral operators $\mathbf{K}_n$ are consistent at u with respect to $\mathbf{K}$.

The Nyström approximation is typically used to derive a low-rank approximation for any kernel machine that can be formulated as a matrix eigenproblem.

2.2.4 Embeddings of Measures

Suppose ν is a probability measure on Ω, which is associated to a probability density $p(x)$. The measure ν can be embedded in $\mathcal{H}$ as

$$\mu[p(x)] := \mathbb{E}[k(\cdot, X)] = \mathbb{E}[\psi(X)], \tag{2.2.26}$$

where X is a random variable distributed according to ν. This embedding, also known as a *mean map*, is a first order statistic of the random variable X in the feature space $\mathcal{H}$[95, 96] and furthermore, allows us to think of probability measures as points in this feature space. The mean map (2.2.26) can be used as a way to measure distances between probability measures, by using the norm of the RKHS $\mathcal{H}$.

Definition 2.2.7. *Let ν and υ be two probability measures on Ω, associated to random variables X and Y. Then the distance measure*

$$\mathrm{MMD}(X,Y) := \|\mathbb{E}[\psi(X)] - \mathbb{E}[\psi(Y)]\|_{\mathcal{H}} \tag{2.2.27}$$

is known as the maximum mean discrepancy (MMD).

What makes this measure useful for machine learning is that the MMD can be computed quite easily for empirical data. For instance, given a sample set $\mathcal{X} = \{x_i\}_1^n$, the empirical mean map is given by

$$\mu[\mathcal{X}] := \frac{1}{n}\sum_{i=1}^{n} k(x_i, \cdot), \tag{2.2.28}$$

which can be used to define an empirical version of (2.2.27). Furthermore, if the kernel used for the mapping has certain properties, the mapping $\mathbb{E}[\psi(\cdot)]$ is *injective*.

Definition 2.2.8. *A continuous kernel k on domain Ω is called* universal *if the space of all functions induced by k is dense in $\mathscr{C}_0(\Omega)$, i.e. for all $f \in \mathscr{C}_0(\Omega)$ and every $\epsilon > 0$, there exists a function g induced by k such that $\|f - g\|_\infty < \epsilon$.*

In this case, the following theorem can be proven.

Theorem 2.2.7. ***(Smola et al [95])*** *If the kernel k is universal, the mean map $\mathbb{E}[\psi(\cdot)]$ is injective.*

This theorem justifies the use of the MMD as a powerful means of discrimination between distributions. Conversely, if two distributions are very close with respect to the distance measure (2.2.27), it implies that one can be used as a proxy for the other, an idea which will be explored in more detail in §3.5.

2.3 Adaptive Control

This section provides a very brief introduction to the paradigm of model-reference adaptive control (MRAC), which will be used as an application of kernel machines to learning time-series data in Chapters 5 and 6. The material in this section primarily follows [38, 42].

2.3.1 Nonlinear Control and Stability

Control theory deals with the design of systems (controllers) that regulate or control a physical process or *plant*. Plants are characterized by reference inputs $r(t)$, outputs

$y(t)$ and control input $u(t)$. The control design task is to choose control inputs $u(t)$ appropriately so that the output satisfies the performance requirements imposed by the engineer. The operation of the plant is approximated by a *model*, which the control design utilizes to create an appropriate controller. A typical mathematical model for a plant is defined by

$$\dot{x}(t) = f(t, x(t), u(t)),\ x(t_0) = x_0 \tag{2.3.1}$$

$$y(t) = h(x(t)), \tag{2.3.2}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, $y \in \mathbb{R}^p$ is the system output, and f is locally Lipschitz continuous in x and u, and piecewise continuous in t. A system state x_e is called an *equilibrium* for (2.3.1) if $f(x(t)) = 0\ \forall t > t_f$, for some t_f. A typical goal of the control input $u(t)$ is to drive a system to a stable equilibrium. Without loss of generality, assume x_e to be 0. Nonlinear systems exhibit different kinds of stability, some of which are defined as follows.

Definition 2.3.1. *(**Stability**) A state $x = 0$ is a*

- Locally stable equilibrium*, if for every $\epsilon > 0$ there exists $\delta(\epsilon) > 0$ such that*

$$\|x(0)\| < \delta(\epsilon) \Rightarrow \|x(t)\| < \epsilon,\ \forall t \geq 0. \tag{2.3.3}$$

- Locally asymptotically stable equilibrium*, if it is stable, and in addition, there exists a $\delta > 0$ such that*

$$\|x(0)\| < \delta \Rightarrow \lim_{t\to\infty} x(t) = 0. \tag{2.3.4}$$

- Locally exponentially stable equilibrium*, if for every $\epsilon > 0$ there exist two strictly positive numbers α and λ such that for some $\delta > 0$*

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \alpha\|x(0)\|e^{-\lambda t},\ \forall t \geq 0. \tag{2.3.5}$$

One can check the stability of equilibria without solving (2.3.1) explicitly by using the Lyapunov stability theorem.

Theorem 2.3.1. *(**Lyapunov** [42]) Let $x = 0$ be an equilibrium point for*

$$\hat{x} = f(x),\ x(0) = x_0,\ x \in \mathbb{R}^n, \tag{2.3.6}$$

in which f is locally Lipschitz in x for $x \in \Omega \subset \mathbb{R}^n$, where Ω contains 0. Suppose that there exists a function $V(x) \in \mathscr{C}^1(\Omega)$ such that $V(0) = 0$ and

$$V(x) > 0\ \forall x \in \Omega \setminus \{0\}$$
$$\dot{V}(x) \leq 0\ x \in \Omega.$$

Then $x = 0$ is stable. Moreover, if

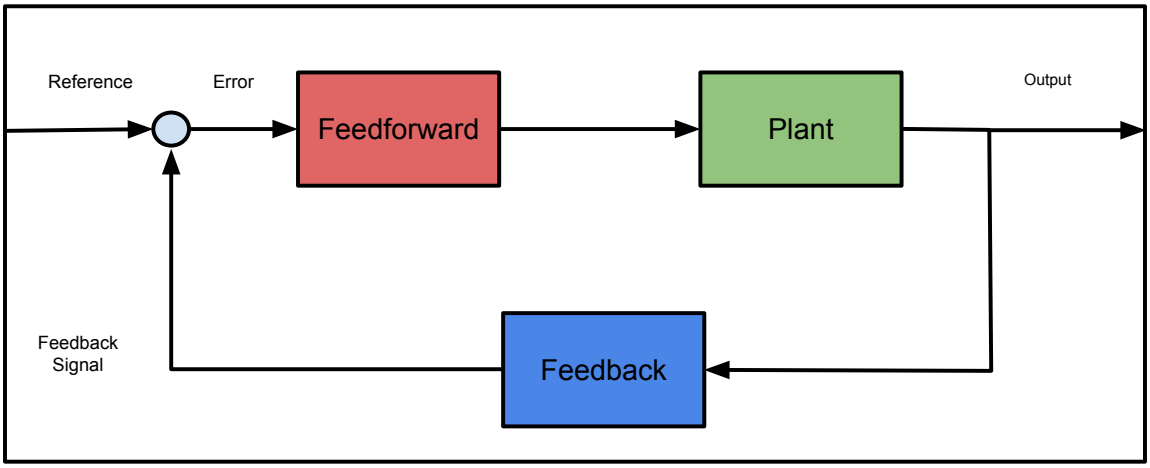
$$\dot{V}(x) < 0,\ x \in \Omega \setminus \{0\},$$

then $x = 0$ is asymptotically stable.

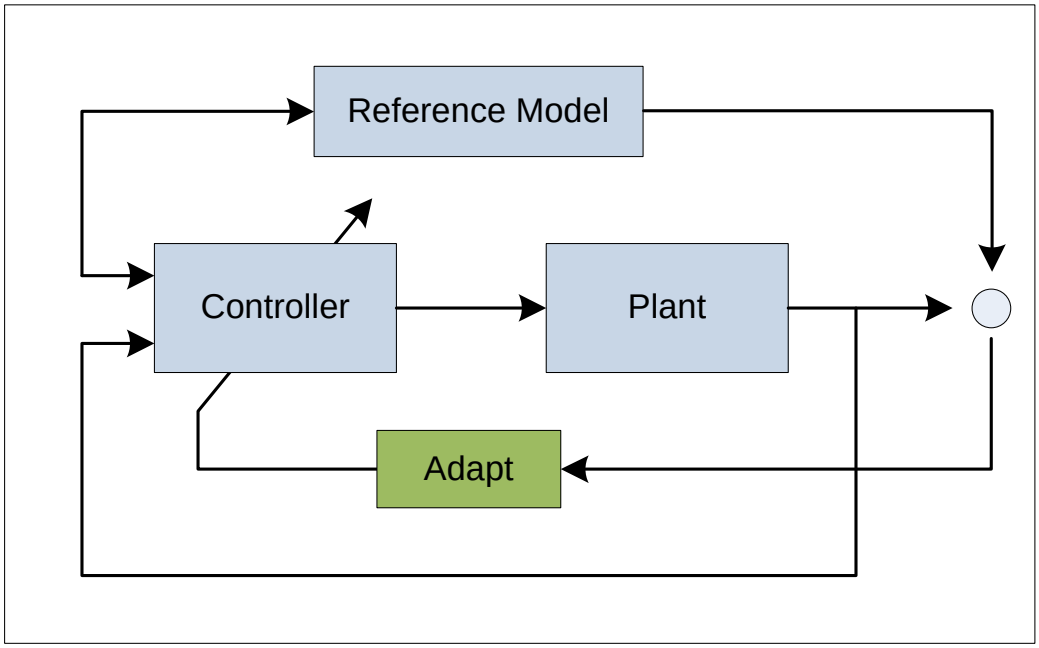
This theorem and related methods can be used to analyze the stability and performance of nonlinear systems.

Feedback control systems are systems in which the control action is dependent on the output. A typical application of a feedback control system is *tracking*, particularly that of a reference input $r(t)$. In this case, the error between the two signals is used as feedback, as shown in Figure 4a. Feedback control systems for other tasks such as stabilization and disturbance rejection can also be designed, and have been studied in great detail [42].

In nonlinear system theory, it is assumed that (2.3.1) is a reasonable model for the system. Typically, however, there is some modeling error associated with (2.3.1), which can have unknown or incorrectly estimated parameters. The goal of adaptive control is to design controllers that can stabilize the system in the face of this error $e(t)$ (termed the uncertainty), and meet the designer's performance specifications. In



(a) A typical feedback control system.



(b) An example of an adaptive control system.

Figure 4

model-reference adaptive control (MRAC), the reference $r(t)$ drives both the plant and a reference model, a model that behaves according to the designer's specifications. The error between these two models is used as feedback in order to make sure (2.3.1)'s response is as ideal as possible. Figure 4b shows an example of the control architecture of MRAC. We will use MRAC as an application for reduced-set model kernel machines, by using them as the adaptation element in MRAC.

PART 1: BATCH DATA

CHAPTER III

REDUCED-SET KERNEL PCA: BASIC THEORY

This chapter outlines a method to speed up both the training and testing phases of certain kernel methods in a unified and principled manner. To do so, the integral operator formulation of kernel PCA is utilized, which leads to a natural framework for computing reduced-set approximations. In particular, a fundamental connection between density estimation and operator approximation is shown, which can be used as a guide for computing the reduced-set models. Experiments on various machine learning applications demonstrate the efficacy of the approach.

3.1 Introduction

Modern problems in machine learning are characterized by large, often redundant, high-dimensional datasets. To interpret and more effectively use high-dimensional data, a simplifying assumption often made is that the data lies on an embedded manifold [55]. Recovery of the underlying manifold aids certain machine learning problems such as deriving a classifier from the data, or estimating a function of interest; many algorithms recovering this underlying structure can be thought of as KPCA performed on specially constructed kernel matrices [35]. In this dissertation, this class of methods is denoted as kernel-manifold-learning algorithms (KMLAs). As noted in the previous section, for a dataset with n points, KMLAs involve the eigendecomposition of an $n \times n$ kernel matrix K, and a manifold mapping of order $\mathcal{O}(n)$ in cost (for a dataset with n points), which limits their usefulness in some application domains (e.g., online learning and visual tracking). In addition to the computational cost, storage of the kernel matrix in memory becomes difficult for larger datasets, particularly for kernels such as the Gaussian, which tends to generate

dense matrices. Therefore a truly scalable KMLA method should be one that 1) avoids the computation of the full kernel matrix, 2) has low training cost, and 3) has low testing cost. These properties are in contrast with methods such as ICD [92] and certain Nyström methods [27], which exhibit excellent performance, but require the computation of the kernel matrix. An example of a Nyström method that does not require the computation of the kernel matrix is one where the centers are chosen uniformly from the data. While performing well in practice, the method suffers from the lack of a principled way to choose the number of centers. Related work in the class is [116], which employs k-means clustering and a density-weighted Gram matrix for performing KPCA. This class of methods requires the retention of the full dataset for computing projections; while the training cost may be lower, the testing cost remains the same. This chapter outlines a method to speed up the training and testing cost associated with KPCA by utilizing the structure of the eigendecomposition of kernel-smoothing operators. In particular, given a sampled dataset $\{x_i\}_1^n$, it is shown that the spectral decomposition of the Gram matrix K is related to the kernel density estimate $\widehat{p}(x)$. If an approximation $\widetilde{p}(x)$ is available whose cardinality is much lower than that of $\widehat{p}(x)$, an approximation to the original Gram matrix can be computed at a significantly reduced computational cost, thus improving the execution of KMLAs.

There are three main contributions made in this research. First, the connection of kernel smoothing to the spectral decomposition of integral operators is exploited, within the context of KPCA, to define reduced-set KPCA (RSKCPA). RSKPCA relies on the existence of a reduced-set density estimate (RSDE) of the dataset, with a cardinality of m rather than n (where $m \ll n$). The RSDE defines a weighted $m \times m$ Gram matrix $\widetilde{K}$, whose eigendecomposition is computed in lieu of the empirical Gram matrix K. The RSKPCA approach circumvents the computation of the full kernel matrix, so that the eigendecomposition is of order $\mathcal{O}(m^3)$ cost, instead of $\mathcal{O}(n^3)$. Evaluation time is also reduced, as mapping a test point into the reduced eigenspace

requires $O(mr)$ operations rather than $O(nr)$, with r retained eigenvectors.

Secondly, for kernels with a bounded maximum, we present results bounding 1) the approximation of the density via the MMD and 2) the difference in Hilbert-Schmidt norm between the operators, providing further theoretical justification for the approach. The two bounds are shown to be directly related, indicating the importance of the density estimate in generating a correct eigendecomposition. Further, the minimization bounds can be used as a guideline for designing RSDEs for good operator approximation.

Finally, while many methods can be used to generate the reduced-set approximation $\widetilde{p}(x)$ to the empirical density $\widehat{p}(x)$, efficient methods are preferred in order to truly impact the overall training time. We present a simple, fast, single-pass method relying on the concept of the 'shadow' of a radially-symmetric kernel to generate the approximation $\widetilde{p}(x)$, called the shadow density estimate (ShDE). The ShDE depends on a user-tuned parameter ℓ to arrive at an RSDE of cardinality $m \ll n$, with a run-time cost of $\mathcal{O}(mn)$. Unlike previous work where m is chosen arbitrarily, ℓ is related to the kernel, and can generally be set to a generic value (say $\ell = 4$) for a wide variety of problems. Also, the error bounds presented earlier can be computed in closed form for ShDE and ShKPCA, in terms of the user-tuned parameter ℓ.

3.2 KPCA and the Nyström Approximation

This section briefly summarizes the foundations of KPCA as regards the spectral decomposition of operators. Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a bounded, positive-definite kernel function, defined on the domain $D \subset \mathbb{R}^d$. The kernel induces a linear operator $\mathbf{K} : \mathcal{L}^2(\Omega) \to \mathcal{L}^2(\Omega)$, as defined in (2.2.12). To incorporate data arising from a probability density $p(x)$, (2.2.12) can be modified. Let ν be a probability measure on Ω associated to p, and denote by $\mathcal{L}^2(\Omega, \nu)$ the space of square-integrable functions with norm $\|f\|_p^2 = \langle f, f \rangle_p = \int_\Omega f(x)^2 d\nu(x)$. Define the linear operator $\mathbf{K} : \mathcal{L}^2(\Omega, \nu) \to$

$\mathcal{L}^2(\Omega, \nu)$ by

$$(\mathbf{K}f)(x) := \int_\Omega k(x,y)f(y)p(y)dy. \tag{3.2.1}$$

The operator $\mathbf{K}$ is associated to the eigenproblem

$$\int_\Omega k(x,y)p(x)\phi_\iota(x)dx = \lambda_\iota \phi_\iota(y), \tag{3.2.2}$$

where $\phi_\iota(\cdot)$ are the eigenfunctions. As noted in [120], if the integral operator (3.2.1) is viewed as an operator on $\mathcal{H}$ (i.e. $\mathbf{K} : \mathcal{H} \to \mathcal{H}$), it is related to the covariance operator by the equation

$$(\mathbf{K}f)(x) := \langle \psi(x), C_\mathcal{H} f \rangle_\mathcal{H}, \tag{3.2.3}$$

where

$$C_\mathcal{H} := \int_\Omega \psi(x) \otimes \psi(x) d\upsilon(x) \tag{3.2.4}$$

and where the tensor product is defined as

$$(\psi(x) \otimes \psi(y))\, f := \langle \psi(y), f \rangle_\mathcal{H} \psi(x), \;\; f \in \mathcal{H}. \tag{3.2.5}$$

For the sake of simplicity, most of the analysis in the chapter takes this viewpoint.

In practice, given a sample set $\mathcal{X} = \{x_1, \ldots, x_n\}$ drawn from $p(x)$, the empirical approximation to (3.2.4) (and hence to (3.2.1)) is given by

$$\widehat{C}_\mathcal{H} := \frac{1}{n} \sum_{i=1}^n \psi(x_i) \otimes \psi(x_i). \tag{3.2.6}$$

Note that $\widehat{C}_\mathcal{H} : \mathcal{H} \to \mathcal{H}$ is a finite-rank operator in infinite-dimensional space. The convergence of $\widehat{C}_\mathcal{H}$ to $C_\mathcal{H}$ has been studied extensively in the asymptotic and non-asymptotic setting [80, 120]. Here, we use the Nyström approximation theorem to show convergence in a very specific sense. To do this, we make note of some assumptions.

Assumption 3.2.1. *The sets of quadrature nodes $\Omega_n \subset \Omega$ in (2.2.21) are sampled i.i.d from ν, while the weights $\alpha_n(x)$ are set to be uniform (i.e. $\alpha_n(x_i) := n^{-1}\ \forall x_i \subset \Omega_n$). Finally, there exists a constant M such that $\|\nu\| \leq M$.*

Note that the approximate measure (2.2.18) then becomes

$$\nu_n(u) = \sum_{x \in \Omega_n} u(x)\alpha_n(x) = \frac{1}{n}\sum_{i=1}^{n} u(x_i). \tag{3.2.7}$$

This approximate measure is consistent with the empirical measure

$$\nu_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}(\Omega) \tag{3.2.8}$$

on the points $x_i \in \Omega_n$. We first need to prove the existence of the finite-sample Nyström extension.

Lemma 3.2.1. *For every $\mathbf{K}_n, u_n\ n \in \mathbb{N}$, there exists an extension $\mathring{\mathbf{K}}_n$ from Ω_n to Ω such that*

$$\left(\mathring{\mathbf{K}}_n u_n\right)(x) = \left\langle \psi(x), \widehat{C}_{\mathcal{H}} u \right\rangle_{\mathcal{H}},\ x \in \Omega, \tag{3.2.9}$$

and $\mathring{\mathbf{K}}_n$ is consistent with $\mathbf{K}_n$ at $x_i \in \Omega_n$.

Proof. This follows from the extension

$$\begin{aligned}
(\mathring{\mathbf{K}}_n u_n)(x) &= \int_{y \in \Omega_n} k_n(x,y) u_n(y) d\upsilon_n(y) \\
&= \frac{1}{n}\sum_{i=1}^{n} k(x, x_i) u(x_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} \langle \psi(x), \psi(x_i)\rangle_{\mathcal{H}} u(x_i) \\
&= \left\langle \psi(x), \frac{1}{n}\sum_{i=1}^{n} \psi(x_i) u(x_i) \right\rangle_{\mathcal{H}} \\
&= \left\langle \psi(x), \frac{1}{n}\sum_{i=1}^{n} \langle \psi(x_i), u\rangle_{\mathcal{H}} \psi(x_i) \right\rangle_{\mathcal{H}} \\
&= \left\langle \psi(x), \left(\frac{1}{n}\sum_{i=1}^{n} \psi(x_i) \otimes \psi(x_i)\right) u \right\rangle_{\mathcal{H}} \\
&= \left\langle \psi(x), \widehat{C}_{\mathcal{H}} u \right\rangle_{\mathcal{H}},
\end{aligned}$$

where we have used the reproducing property in the second-to-last line. Evaluating $(\mathring{\mathbf{K}}_n u_n)(x)$ on $x_i \in \Omega_n$ shows consistency with $\mathbf{K}_n$, proving the theorem. □

We can now prove the asymptotic convergence of $\widehat{C}_{\mathcal{H}}$ to $C_{\mathcal{H}}$.

Theorem 3.2.1. *Let the assumptions in Assumption 3.2.1 hold. Then*

$$\lim_{n\to\infty} \|\widehat{C}_{\mathcal{H}} - C_{\mathcal{H}}\|_{f,\Omega_n} \to 0, \ \forall f \in \mathcal{H}, \ \Omega_n \subset \Omega, \tag{3.2.10}$$

where

$$\|\widehat{C}_{\mathcal{H}} - C_{\mathcal{H}}\|_{f,\Omega_n} := \sup_{x\in\Omega_n} \left\| \left\langle \psi(x), \left(\widehat{C}_{\mathcal{H}} - C_{\mathcal{H}}\right) f \right\rangle_{\mathcal{H}} \right\|, \ f \in \mathcal{H}.$$

Proof. For the Nyström method, we have from (3.2.3) and Lemma 3.2.1 that

$$\begin{aligned} d_n^k(f)(x) &= \sum_{y\in\Omega_n} \alpha_n(y)k(x,y)f(y) - \int_\Omega k(x,y)f(y)d\nu(y) \\ &= \left\langle \psi(x), \widehat{C}_{\mathcal{H}} f \right\rangle_{\mathcal{H}} - \langle \psi(x), C_{\mathcal{H}} f \rangle_{\mathcal{H}} \\ &= \left\langle \psi(x), \left(\widehat{C}_{\mathcal{H}} - C_{\mathcal{H}}\right) f \right\rangle_{\mathcal{H}}. \end{aligned}$$

By the strong law of large numbers, the empirical measure (3.2.8) converges almost surely to ν; therefore, we have that $\nu_n \rightharpoonup \nu$. Furthermore, since $\|\nu_n\|$ is finite, $\exists M'$ s.t. $\|\nu_n\| + \|\nu\| \leq M'$. Applying the Nyström approximation theorem then proves the result. □

The empirical integral operator (3.2.9) generated by the Nyström extension is the object utilized in the empirical KPCA procedure, whose goal is to approximate the feature map (2.2.14). The feature map is generated by a solution of the eigenproblem (3.2.2); therefore KPCA requires the solution of the approximate eigenproblem

$$\widehat{C}_{\mathcal{H}}\widehat{\phi}_\iota = n\widehat{\lambda}_\iota\widehat{\phi}_\iota. \tag{3.2.11}$$

The approximate eigenfunctions can be written as

$$\widehat{\phi}_\iota = \sum_{i=1}^n u_{\iota_i}\psi(x_i). \tag{3.2.12}$$

The solution to (3.2.11) is given by

$$K^{(n)}U^{(n)} = U^{(n)}\Lambda^{(n)}, \tag{3.2.13}$$

where $K^{(n)} \in \mathbb{R}^{n\times n}$, $K^{(n)}_{ij} := k(x_i, x_j)$ and the ιth column of $U^{(n)}$ is u_ι [91]. In the KPCA literature, it is required that the eigenfunctions be unit norm, which can be enforced by requiring $n\lambda u_\iota^T u_\iota = 1$ for all u_ι. Therefore, this procedure incudes the empirical kernel map

$$\widehat{\psi}_{\text{kpca}}(x) = n\left(\sum_{j=1}^{n} k(x, x_j)U^{(n)}_{j,1}, \sum_{k=1}^{n} k(x, x_j)U^{(n)}_{j,2}, \ldots, \sum_{j=1}^{n} k(x, x_j)U^{(n)}_{j,n}\right). \tag{3.2.14}$$

From construction, it's clear that the training cost is $\mathcal{O}(n^3)$ and the testing cost is $\mathcal{O}(rn)$, where r is the rank of $\widehat{\psi}_{\text{kpca}}$ desired.

In order to get a low-rank approximation to $\widehat{\psi}_{\text{kpca}}$ with significantly reduced training complexity, the Nyström procedure can be applied again to a reduced set $\mathrm{C} := \{c_1, \ldots, c_m\} \subset \{x_1, \ldots, x_n\}$, where $m < n$ [113]. After choosing the subset C, the following eigenproblem is solved:

$$K^{(m)}U^{(m)} = U^{(m)}\Lambda^{(m)}, \tag{3.2.15}$$

where $K^{(m)} \in \mathbb{R}^{m\times m}$ and $K^{(m)}_{ij} := k(c_i, c_j)$. A low-rank approximation of the eigenvectors of the original kernel matrix can be computed as

$$\grave{U}^{(n)}_{j,i} = \sqrt{\frac{m}{n}}\frac{1}{\lambda^{(m)}_i}K^{(n,m)}U^{(m)}_i, \tag{3.2.16}$$

and

$$\grave{\lambda}_i = \frac{n}{m}\lambda^{(m)}_i. \tag{3.2.17}$$

where $U^{(m)}_i$ is the ith column of $U^{(m)}$, and $K^{(n,m)} \in \mathbb{R}^{n\times m}$ is the extrapolation matrix defined as $K^{(n,m)}_{ij} := k(x_i, c_j)$. This procedure induces the empirical kernel map

$$\widehat{\psi}_{\text{ny}}(x) = n\left(\sum_{j=1}^{n} k(x, x_j)\grave{U}^{(n)}_{j,1}, \sum_{k=1}^{n} k(x, x_j)\grave{U}^{(n)}_{j,2}, \ldots, \sum_{j=1}^{n} k(x, x_j)\grave{U}^{(n)}_{j,r}\right). \tag{3.2.18}$$

Here, the training cost is $\mathcal{O}(m^3)$ and the testing cost is $\mathcal{O}(rn)$.

3.3 Reduced-Set KPCA

This section proposes an alternative formulation of the operator and its spectral decomposition, to derive reduced-set KPCA, as based on an approximation to the empirically determined kernel density estimate.

First, note that the KPCA integral equation (3.2.1) applied to the indicator function $\mathbf{1}(x)$ implies a kernel *smoothing* of the density. Given a set of samples $\mathcal{X} = \{x_1, \ldots, x_n\}$ drawn from the measure ν, and using (3.2.8), the smoothed approximation is obtained as

$$\widehat{p}(x) = (\mathring{\mathbf{K}}_n \mathbf{1})(x) = \frac{1}{n} \sum_{i=1}^{n} k(x_i, x), \tag{3.3.1}$$

which, for certain kernels such as the Gaussian, is known as the *kernel density estimate (KDE)* of $p(x)$ [110]. It is important at this point to make a distinction between the classical theory of KDEs and the estimate (3.3.1). In nonparametric statistics, it is assumed that the kernel integrates to one, and has a free parameter, known as a *bandwidth*, which controls its shape. For instance, the Gaussian kernel (2.2.9) with normalization constant C_g and bandwidth $2\sigma^2$ is a valid kernel for a KDE. In order for the KDE to converge to the true density $p(x)$ in mean-squared error, the bandwidth must decrease as a function of the samples [110]. On the other hand, the free parameter is *independent* of the samples in kernel methods, because inference is performed in a *fixed* feature space $\mathcal{H}$. Therefore, it is possible that the structure of the kernel induces a large amount of smoothing upon the density, rendering a lot of the data redundant, and thus amenable to approximation.

One of the limiting factors of using KDEs in practice is the $\mathcal{O}(n)$ number of operations required to compute $\widehat{p}(x)$. It is therefore common to utilize a *reduced-set density estimate*

$$\widetilde{p}(x) = \frac{1}{n} \sum_{i=1}^{m} w_i k(c_i, x), \tag{3.3.2}$$

where $\mathcal{W} = \{w_1, \ldots, w_m\}$, $\mathrm{C} = \{c_1, \ldots, c_m\}$, and $m \ll n$. The empirical measure

generating $\widetilde{p}$ under the kernel smoother $\mathbf{K}$ is

$$\widehat{\nu}_n = \frac{1}{n}\sum_{i=1}^{m} w_i \delta_{c_i}. \tag{3.3.3}$$

While having quite different generating approximations, the kernel-smoothed density $\widetilde{p}$ is close to $\widehat{p}$ by construction [15, 31, 116]. This work replaces the KPCA procedure of the eigenproblem derived from (3.2.11) and (3.2.8) with one derived from (3.3.3).

Let the reduced-set centers be given by $\mathrm{C} = \{c_1, \ldots, c_m\}$, and define the data-to-center mapping $\vartheta : \{1, \ldots, n\} \to \{1, \ldots, m\}$. The set of centers C and the mapping ϑ can be used to generate the empirical measure (3.3.3), and allows us to prove the following theorem.

Theorem 3.3.1. *Let a mapping ϑ, defined as above, be available. Then the solution to the eigenproblem associated to the quantized empirical covariance operator*

$$\widetilde{C}_{\mathcal{H}} = \frac{1}{n}\sum_{i=1}^{n} \psi(c_{\vartheta(i)}) \otimes \psi(c_{\vartheta(i)}) \tag{3.3.4}$$

can be computed by the eigendecomposition of the matrix

$$n\widetilde{\lambda}_\iota W^{\frac{1}{2}} \alpha_\iota = W^{\frac{1}{2}} \overline{K}^2 W^{\frac{1}{2}} \left(W^{\frac{1}{2}} \alpha_\iota \right), \tag{3.3.5}$$

where $\overline{K}_{ij} := k(c_i, c_j)$ and $W_{ii} := w_i$ is a diagonal matrix of weights.

Proof. From the representer theorem, the eigenfunctions of the operator (3.3.4) have the expansion

$$\begin{aligned} \widetilde{\phi}_\iota &= \sum_{i=1}^{n} \alpha_{\iota_i} \psi(c_{\vartheta(i)}) \\ &= \sum_{i=1}^{m} \alpha_{\iota_i} w_i \psi(c_i). \end{aligned} \tag{3.3.6}$$

Projecting one of the basis elements $\psi(c_i)$ onto the eigenfunction yields

$$\begin{aligned} \left\langle \psi(c_i), \widetilde{\phi}_\iota \right\rangle_{\mathcal{H}} &= \left\langle \psi(c_i), \sum_{j=1}^{m} \alpha_{\iota_j} w_j \psi(c_j) \right\rangle_{\mathcal{H}} \\ &= \sum_{j=1}^{m} \alpha_{\iota_j} w_j k(c_i, c_j). \end{aligned}$$

This computation can be utilized for the eigenfunction equation

$$\begin{aligned}
\widetilde{C}_{\mathcal{H}}\widetilde{\phi}_\iota &= \frac{1}{n}\left(\sum_{i=1}^{n}\psi(c_{\vartheta(i)})\otimes\psi(c_{\vartheta(i)})\right)\widetilde{\phi}_\iota \\
&= \frac{1}{n}\sum_{i=1}^{n}\left\langle\psi(c_{\vartheta(i)}),\widetilde{\phi}_\iota\right\rangle_{\mathcal{H}}\psi(c_{\vartheta(i)}) \\
&= \frac{1}{n}\sum_{i=1}^{m}w_i\left\langle\psi(c_i),\widetilde{\phi}_\iota\right\rangle_{\mathcal{H}}\psi(c_i) \\
&= \frac{1}{n}\sum_{i=1}^{m}w_i\underbrace{\left(\sum_{j=1}^{m}\alpha_{\iota_j}w_jk(c_i,c_j)\right)}_{\beta_{\iota_i}}\psi(c_i) \\
&= \frac{1}{n}\sum_{i=1}^{m}w_i\beta_{\iota_i}\psi(c_i),
\end{aligned}$$

Projecting the bases onto $\widetilde{\lambda}_\iota\widetilde{\phi}_\iota$ gives

$$\begin{aligned}
\left\langle\psi(c_q),\widetilde{\lambda}_\iota\widetilde{\phi}_\iota\right\rangle_{\mathcal{H}} &= \widetilde{\lambda}_\iota\langle\psi(c_q),\widetilde{\phi}_\iota\rangle_{\mathcal{H}} \\
&= \widetilde{\lambda}_\iota\sum_{i=1}^{m}\alpha_{\iota_i}w_ik(c_q,c_i),
\end{aligned}$$

whereas the projection of the same basis onto $\widetilde{C}_{\mathcal{H}}\widetilde{\phi}_\iota$ gives

$$\begin{aligned}
\left\langle\psi(c_q),\widetilde{C}_{\mathcal{H}}\widetilde{\phi}_\iota\right\rangle_{\mathcal{H}} &= \left\langle\psi(c_q),\frac{1}{n}\sum_{i=1}^{m}w_i\beta_{\iota_i}\psi(c_i)\right\rangle_{\mathcal{H}} \\
&= \frac{1}{n}\sum_{i=1}^{m}w_i\beta_{\iota_i}k(c_q,c_i) \\
&= \frac{1}{n}\sum_{i=1}^{m}w_i\left(\sum_{j=1}^{m}\alpha_{\iota_j}w_jk(c_i,c_j)\right)k(c_q,c_i) \\
&= \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{m}w_iw_j\alpha_{\iota_j}k(c_i,c_j)k(c_q,c_i).
\end{aligned}$$

These equations can be written as

$$\left\langle\psi(c_q),\widetilde{\lambda}_\iota\widetilde{\phi}_\iota\right\rangle_{\mathcal{H}} = \widetilde{\lambda}_\iota W\overline{K} \tag{3.3.7}$$

and

$$\left\langle\psi(c_q),\widetilde{C}_{\mathcal{H}}\widetilde{\phi}_\iota\right\rangle_{\mathcal{H}} = \frac{1}{n}W\overline{K}\overline{K}W, \tag{3.3.8}$$

where $\overline{K}_{ij} := k(c_i, c_j)$ and $W_{ii} := w_i$ is a diagonal matrix of weights. Equating these leads to the matrix eigenproblem

$$\begin{aligned} n\widetilde{\lambda}_\iota W\overline{K}\alpha_\iota &= W\overline{K}^2 W\alpha_\iota \\ n\widetilde{\lambda}_\iota\alpha_\iota &= \overline{K}W\alpha_\iota, \end{aligned} \tag{3.3.9}$$

since k is positive-definite. Finally, (3.3.9) can be made symmetric by the transform

$$n\widetilde{\lambda}_\iota W^{\frac{1}{2}}\alpha_\iota = W^{\frac{1}{2}}\overline{K}W^{\frac{1}{2}}\left(W^{\frac{1}{2}}\alpha_\iota\right), \tag{3.3.10}$$

which leads to a more stable computation, and proves the theorem. □

This theorem results in Algorithm 1, which we call *reduced-set KPCA* (RSKPCA). The key thing to note is that the proposed RSKPCA procedure replaces the Gram matrix K in the empirical eigenproblem (3.2.11) by a density weighted surrogate $\widetilde{K} = WK^{\mathrm{C}}W^T$, where $K^{\mathrm{C}}_{ij} := k(c_i, c_j)$, and $W = \mathrm{diag}(\sqrt{w_1}, \ldots, \sqrt{w_m})$ is the weight matrix. Unlike K, K^{C} is an $m \times m$ matrix (as is $\widetilde{K}$). Once the centers are selected, and the weights computed using a reduced-set density estimation algorithm, *the original data is discarded*. Due to the removal of the data, the algorithm is different from Nyström methods, which retain the training data for eigenfunction computations at test time. The algorithm is also different from sparse approximation and eigenvector approximation methods, which need to compute the eigendecomposition of the full kernel matrix to generate the reduced-set eigenfunction computations for testing. RSKPCA can be more aggressive with the training data than either of these two strategies in pursuit of both training and testing speedups, and is summarized in Algorithm 1. Since the full kernel matrix is never computed once an RSDE is available, the training cost of the algorithm is $\mathcal{O}(m^3)$, and the testing cost is $\mathcal{O}(m)$.

3.4 The Role of Density-Weighting

This section gives an empirical demonstration of the role the density $p(x)$ plays in the approximation of the eigenfunctions $\phi(x)$ of the operator (2.2.12).

Algorithm 1 Reduced-Set KPCA

Input: Initial data $\mathcal{X} = \{x_1, \ldots, x_n\}$, $x_i \in \mathbb{R}^d$.

Procedure:

1) Apply a reduced-set density estimator to $\mathcal{X}$ to compute $\mathrm{C} = \{c_1, \ldots, c_m\}$ and $w = \{w_1, \ldots, w_m\}$.

2) Create diagonal matrix $W = \mathrm{diag}(\sqrt{w_1}, \ldots, \sqrt{w_m})$.

3) Compute weighted kernel matrix

$$\widetilde{K} \in \mathbb{R}^{m \times m}, \quad \widetilde{K} := W K^{\mathrm{C}} W$$

where $K^{\mathrm{C}}_{ij} := k(c_i, c_j)$.

4) Perform eigenvector decomposition $\widetilde{K}\alpha_\iota = \lambda_\iota \alpha_\iota$.

5) Reweight to get the eigenvectors $\alpha_\iota \mapsto W^{1/2}\alpha_\iota$.

Output:

Compute up to m eigenfunctions as

$$\widetilde{\phi}_\iota(x) = \sum_{i=1}^{m} \alpha_{\iota_i} k(c_i, x). \tag{3.3.11}$$

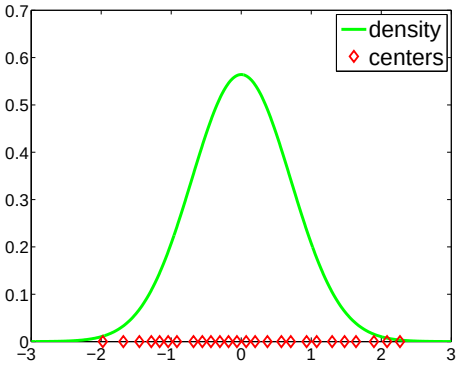
3.4.1 A Toy Example

In Algorithm 1, the coefficients α_ι associated to the approximate eigenfunction $\widetilde{\phi}_\iota(z)$ are re-weighted by the weight matrix W. One of the hypotheses behind RSKPCA is that until the reweighting is applied, the eigenfunctions $\widetilde{\phi}_\iota$ are not scaled appropriately. In order to test whether this hypothesis truly holds, consider the following analytic case [112, 116]:

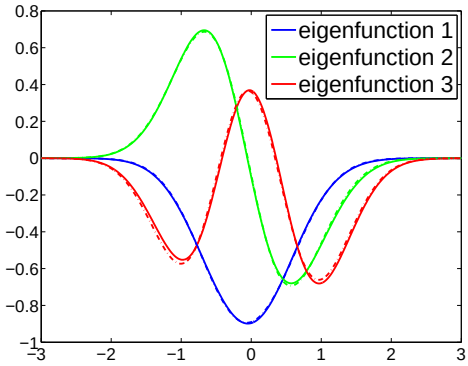
1. $p(x) = \sqrt{\frac{2a}{\pi}} e^{-2ax^2}$.
2. $k(x, y) = e^{-b(x-y)^2}$.

The parameters $a = 1$ and $b = 3$ give a Gaussian density with variance $1/2$ and a kernel with bandwidth $\sigma^2 = 1/6$. 600 samples were generated from the density, and KPCA, the weighted RSKPCA and the unweighted RSKPCA were run on the samples. The results are shown in Figure 5. These results provide experimental illustration that the weights play a role in RSKPCA, because their removal impacts

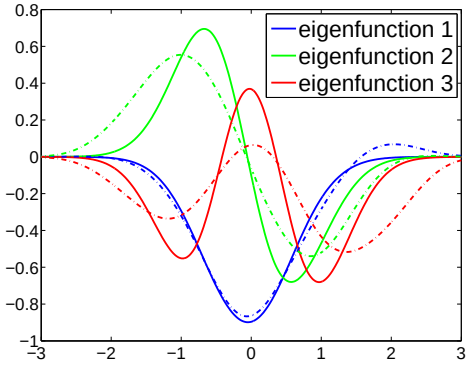
the fidelity of the approximate eigenfunctions to those of KPCA.



(a) Probability density with centers



(b) Weighted eigenfunction approximation



(c) Unweighted eigenfunction approximation

Figure 5: Theoretical example of eigenfunction approximation. The solid blue, green and red lines represent the eigenfunctions computed by KPCA, whereas the dashed lines represent the weighted and unweighted RSKPCA approximations.

3.4.2 Real World Data

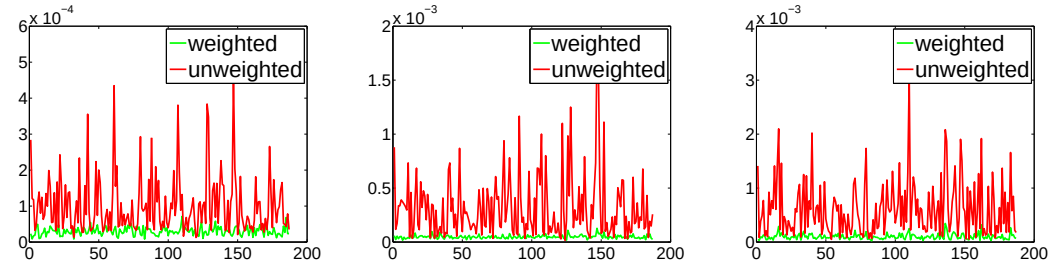
Similar results hold on datasets arising from more complicated distributions in higher dimensions. Consider the **german** dataset, which has $1,000$ samples in 24 dimensions. Note that in this case, the problem of checking the eigenfunction approximation becomes more difficult, since we have a 24 dimensional space over which to check each function. To make things easier, the following approach was used on weighted and unweighted RSKPCA:

1. Train on 80% of the data.
2. Compute embeddings on the remaining 20% of the data.
3. Average the error of the embeddings with respect to the KPCA embedding.

Results for the **german** dataset with $\sigma = 30$ are given in Figure 6. The data retention rate for RSKPCA here is about 10%. The **dna** dataset is also considered, which has $2,000$ samples in 180 dimensions. Here, with $\sigma = 23.5$, results similar to the **german** dataset are achieved, as shown in Figure 7. From these figures, it's clear that there is empirical evidence suggesting that reweighting the eigenfunctions is an appropriate strategy to truly learn a empirical kernel map similar to the ideal eigenfunctions $\phi_\iota(x)$. The next section focuses on demonstrating the well-foundedness of the approach by bounding the approximation error between the operators induced by the algorithms in a few different metrics.

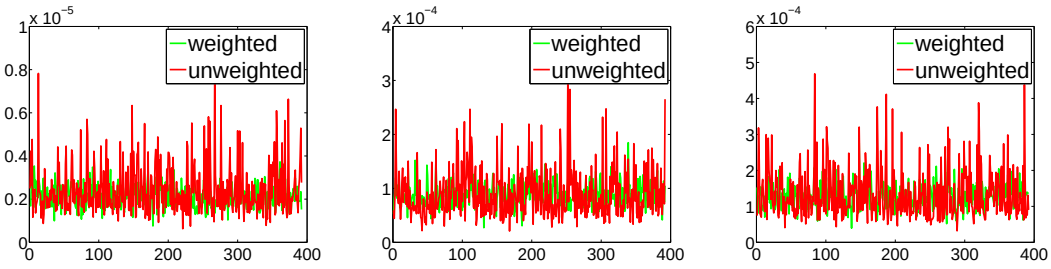
3.5 Analysis of Approximation Error

This section reports bounds on the MMD error for RSDEs, as well as bounds on the difference between the eigenvalues and spectral projections of the operators associated to the original kernel matrix generated by KPCA, K, and the one generated by the RSDE, $\widetilde{K}$. The bounds demonstrate the claim that an accurate RSDE leads to an accurate eigendecomposition, since the bounds on the approximation error of



(a) First component error (b) Second component error (c) Third component error

Figure 6: Embedding error on **german** dataset.



(a) First component error (b) Second component error (c) Third component error

Figure 7: Embedding error on **dna** dataset.

the eigendecomposition are given in terms of the error of the approximated density estimate.

As stated in §2.2.1, the MMD is a distance measure between probability measures in the Hilbert space $\mathcal{H}$ induced by the kernel k [95]. The (biased) empirical MMD is defined to be

$$\mathrm{MMD}(\mathcal{X},\mathcal{Y})_b^2 := \left\| \sum_{i=1}^{n} \frac{1}{n}\psi(x_i) - \sum_{i=1}^{n} \frac{1}{n}\psi(y_i) \right\|_{\mathcal{H}}^2, \tag{3.5.1}$$

where the b denotes bias, and ψ is the mapping from the input space $\mathbb{R}^d$ to $\mathcal{H}$. The points $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are generated by probability measure ν and υ respectively; both sets have the same number of elements. Since the kernel in KPCA induces a smoothing effect on the samples from the true probability density p, a small value for the MMD between the KDE and an RSDE is indicative of the RSDE acting as an effective surrogate for p in the KPCA space, thus generating an effective approximation

to (3.2.6) via the use of Algorithm 1.

In the analysis below, it is assumed that the kernel k has a maximum value attained at $k(x, x_{\max_j}) \leq \kappa$. Define the sets $\mathcal{X}$ and C, and the data-to-center mapping $\vartheta : \{1, \ldots, n\} \to \{1, \ldots, m\}$ as above. Then the following theorem upper bounds the difference in MMD between $\widehat{p}(x)$ and $\widetilde{p}(x)$.

Proposition 3.5.1. ***(MMD Worst Case Bound 1)*** *Let n be the number of samples, $\mathcal{X}$, C and ϑ be defined as above. Then*

$$\mathrm{MMD}(X, \mathrm{C})_b \leq \sum_{j=1}^{m} \sqrt{2} w_j \sqrt{\left(\kappa - k(x_{\max_j}, c_j)\right)}, \tag{3.5.2}$$

where κ is the maximum value of the kernel, $x_{\max_j} = \arg\min_{x_i \in S_j} k(x_i, c_j)$, and S_j represents the set of points assigned to center c_j.

Proof. The MMD can be broken up into subproblems

$$\begin{aligned}
\mathrm{MMD}_b(X, \bar{\mathrm{C}}) &= \left\| \sum_{i=1}^{n} \frac{1}{n} \left(\psi(x_i) - \psi(c_{\vartheta(i)}) \right) \right\|_{\mathcal{H}} \\
&= \frac{1}{n} \left\| \sum_{j=1}^{m} \sum_{x_i \in S_j} \left(\psi(x_i) - \psi(c_j) \right) \right\|_{\mathcal{H}} \\
&\leq \sum_{j=1}^{m} \frac{1}{n} \left\| \sum_{x_i \in S_j} \left(\psi(x_i) - \psi(c_j) \right) \right\|_{\mathcal{H}},
\end{aligned}$$

due to the properties of the norm. Therefore the following quantities need to be bounded:

$$\begin{aligned}
\mathrm{MMD}_{\mathrm{sub}_j} &= \left\| \sum_{x_i \in S_j} \left(\psi(x_i) - \psi(c_j) \right) \right\|_{\mathcal{H}} \\
&\leq \sum_{x_i \in S_j} \left\| \left(\psi(x_i) - \psi(c_j) \right) \right\|_{\mathcal{H}} \\
&\leq \sum_{x_i \in S_j} \left\| \epsilon_i \right\|_{\mathcal{H}}.
\end{aligned}$$

Now

$$\begin{aligned}\|\epsilon_i\|_{\mathcal{H}}^2 &= \langle \epsilon_i, \epsilon_i \rangle_{\mathcal{H}} \\ &= \langle \psi(x_i) - \psi(c_{\vartheta(i)}), \psi(x_i) - \psi(c_{\vartheta(i)}) \rangle_{\mathcal{H}} \\ &= 2\left(\kappa - k(x_i, c_{\vartheta(i)}\right) \\ &\leq 2\left(\kappa - k(x_{\max_j}, c_j)\right),\end{aligned}$$

where $x_{\max_j}$ is the boundary point maximizing the above expression. Taking the square root, we get

$$\|\epsilon_i\|_{\mathcal{H}} \leq \sqrt{2\left(\kappa - k(x_{\max_j}, c_j)\right)},$$

Using the fact that $w_j := |S_j|$ and summing over all the subproblems, proves the proposition. □

Corollary 3.5.1. *For the minimization problem associated to minimizing (3.5.1), an upper bound on* $\mathrm{MMD}(X, \mathrm{C})_b$ *corresponds to*

$$\min_{S_j \subset \mathcal{X}, c_j \in D} \sum_{j=1}^{m} \sum_{x_i \in S_j} \sqrt{\left(\kappa - k(x_i, c_{\vartheta(i)})\right)}. \tag{3.5.3}$$

We now show that the distance in Hilbert-Schmidt norm between the empirical operators generated by the KPCA and RSKPCA procedures is bounded. Using the notation of [80], let $\psi(x_i) = k_{x_i} := k(\cdot, x_i)$; therefore,

$$\widehat{C}_{\mathcal{H}} := \frac{1}{n} \sum_{i=1}^{n} \langle \cdot, k_{x_i} \rangle_{\mathcal{H}} k_{x_i}, \tag{3.5.4}$$

where $\langle \cdot, k_{x_i} \rangle_{\mathcal{H}}$ projects the point onto the kernel function k_{x_i}. The definition of $\widetilde{C}_{\mathcal{H}}$ follows similarly. This leads to the following proposition.

Theorem 3.5.1. ***(Hilbert-Schmidt Worst Case Bound 1)*** *Let* $\widehat{C}_{\mathcal{H}}$ *and* $\widetilde{C}_{\mathcal{H}}$ *be defined as above. Then*

$$\left\| \widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} \right\|_{\mathrm{HS}} \leq \frac{2\sqrt{\kappa}}{n} \sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(x_{max_j}, c_j)\right)}. \tag{3.5.5}$$

Proof. Since $k_{x_i} \in \mathcal{H}$, define a residual in the space as $\epsilon_i := k_{x_i} - k_{c_{\vartheta(i)}}$. Then

$$\begin{aligned}
\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} &= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} k_{x_i} - \langle\cdot, k_{c_{\vartheta(i)}}\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} k_{x_i} - \langle\cdot, k_{x_i} - \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} k_{x_i} - \langle\cdot, k_{x_i}\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}} + \langle\cdot, \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} \left(k_{x_i} - k_{c_{\vartheta(i)}}\right) + \langle\cdot, \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} \epsilon_i + \langle\cdot, \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right).
\end{aligned}$$

Therefore

$$\begin{aligned}
\left\|\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right\|_{\mathrm{HS}} &\leq \left\|\frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} \epsilon_i + \langle\cdot, \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right)\right\|_{\mathrm{HS}} \\
&\leq \left\|\frac{1}{n}\sum_{i=1}^{n}\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} \epsilon_i\right\|_{\mathrm{HS}} + \left\|\frac{1}{n}\sum_{i=1}^{n}\langle\cdot, \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right\|_{\mathrm{HS}}.
\end{aligned}$$

As before, divide the data into sets $S_j,\ j = 1\ldots, m$. Then, the above inequality is further bounded by

$$\left\|\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right\|_{\mathrm{HS}} \leq \frac{1}{n}\left(\left\|\sum_{j=1}^{m}\sum_{x_i\in S_j}\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} \epsilon_i\right\|_{\mathrm{HS}} + \left\|\sum_{j=1}^{m}\sum_{x_i\in S_j}\langle\cdot, \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right\|_{\mathrm{HS}}\right).$$

Using the definition of the tensor product, we have

$$\begin{aligned}
\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} \epsilon_i &= \psi(x_i) \otimes \epsilon_i \\
\langle\cdot, \epsilon_i\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}} &= \epsilon_i \otimes \psi(c_{\vartheta(i)}).
\end{aligned}$$

From [34],

$$\begin{aligned}
\|\psi(x_i) \otimes \epsilon_i\|_{\mathrm{HS}}^2 &= \langle\psi(x_i), \psi(x_i)\rangle_{\mathcal{H}}\langle\epsilon_i, \epsilon_i\rangle_{\mathcal{H}} \leq \kappa\|\epsilon_i\|_{\mathcal{H}}^2, \\
\left\|\epsilon_i \otimes \psi(c_{\vartheta(i)})\right\|_{\mathrm{HS}}^2 &= \langle\epsilon_i, \epsilon_i\rangle_{\mathcal{H}}\langle\psi(c_{\vartheta(i)}), \psi(c_{\vartheta(i)})\rangle_{\mathcal{H}} \leq \kappa\|\epsilon_i\|_{\mathcal{H}}^2.
\end{aligned}$$

From repeated applications of the triangle inequality, we have

$$\begin{aligned}\left\|\widehat{C}_{\mathcal{H}}-\widetilde{C}_{\mathcal{H}}\right\|_{\mathrm{HS}} &\leq \frac{1}{n}\left(\left\|\sum_{j=1}^{m}\sum_{x_i\in S_j}\langle\cdot,k_{x_i}\rangle_{\mathcal{H}}\epsilon_i\right\|_{\mathrm{HS}}+\left\|\sum_{j=1}^{m}\sum_{x_i\in S_j}\langle\cdot,\epsilon_i\rangle_{\mathcal{H}}k_{c_{\vartheta(i)}}\right\|_{\mathrm{HS}}\right)\\ &\leq \frac{1}{n}\sum_{j=1}^{m}\sum_{x_i\in S_j}\left(\left\|\langle\cdot,k_{x_i}\rangle_{\mathcal{H}}\epsilon_i\right\|_{\mathrm{HS}}+\left\|\langle\cdot,\epsilon_i\rangle_{\mathcal{H}}k_{c_{\vartheta(i)}}\right\|_{\mathrm{HS}}\right)\\ &\leq \frac{1}{n}\sum_{j=1}^{m}\sum_{x_i\in S_j}2\sqrt{\kappa}\|\epsilon_i\|_{\mathcal{H}}.\end{aligned}$$

Recall $\epsilon_i=\psi(x_i)-\psi(c_{\vartheta(i)})$. Substituting, we get

$$\left\|\widehat{C}_{\mathcal{H}}-\widetilde{C}_{\mathcal{H}}\right\|_{\mathrm{HS}}\leq\frac{1}{n}\sum_{j=1}^{m}w_j\sqrt{2\left(\kappa-k(x_{\max_j},c_j)\right)}.$$

□

Corollary 3.5.2. *The minimization problem associated to minimizing (3.5.5), an upper bound on* $\left\|\widehat{C}_{\mathcal{H}}-\widetilde{C}_{\mathcal{H}}\right\|_{\mathrm{HS}}$*, corresponds to (3.5.3).*

The minimization problem (3.5.3) can be used as a guide to develop an algorithm to compute an affect RSDE for RSKPCA. If, for example, the kernel is Gaussian, a gradient-based method similar to k-means can be used; see Appendix A for a derivation of this method.

3.6 Shadow Densities: a Fast and Simple RSDE

Here, a specific RSDE algorithm for use within RSKPCA, to improve the execution time of learning and testing versus KPCA, is given. By proposing a simple algorithm, closed-form approximation errors are computable, as explored in the subsequent section.

While many algorithms have been designed for reduced set density estimation, to meet our purposes, the RSDE must satisfy three criteria: 1) it must incorporate the kernel within its estimate; 2) its computational cost cannot be excessive, as that would fail to speed up the KMLA; and 3) the number of centers m must be identified

in a principled way, since they may vary from problem to problem, and must have deterministic approximation error. These three criteria are met by a simple algorithm exploiting the structure of radially symmetric kernels. An approach similar to the one proposed here is found in [109], however their selection parameter is not fundamentally related to the kernel bandwidth and they draw no connection to KPCA.

Consider a bounded kernel function $k(\cdot,\cdot)$, where κ is the maximum value attained at $k(c,c)\ \forall c \in \mathbb{R}^d$, and a sequence $\{y_i\}_{i\in\mathbb{N}}$. If $\|c - y_i\| \to 0$, then $k(c,y_i) \to \kappa$ (as $i \to \infty$). Points sufficiently close to c seem indistinguishable from the perspective of the kernel centered at c. Declare such points near c to lie in the *shadow* of the kernel function at c. Given a dataset $\{x_i\}_1^n$ used to determine $\widehat{p}(x)$, all points of the dataset in the shadow of another point $c \in \{x_i\}_1^n$ can be replaced with c at minor cost, leading to the shadow density estimate:

$$\widetilde{p}(x) := \frac{1}{n}\sum_{j=1}^{m} w_j k(c_j, x) \approx \frac{1}{n}\sum_{j=1}^{m}\sum_{\xi\in S_j} k(\xi, x), \tag{3.6.1}$$

where S_j is the set of points lying in the shadow of the point c_j, $w_j = |S_j|$, and $S_i \cap S_j = \emptyset$ when $i \neq j$. This proposal specializes to the case of radially symmetric kernels with bandwidth parameter σ, and defines ε to be determined by a parameter ℓ via $\varepsilon(\ell) = \sigma/\ell$. What remains is to provide a selection procedure for the shadow centers c_j. Algorithm 2 provides a single-pass, $\mathcal{O}(mn)$ complexity approach. Figure 8 conceptually depicts the process of moving from data to shadow centers, and also the reconstruction of the KDE using a ShDE. The color coding depicts the distinct shadow sets. Based on §3.2, the RSKPCA procedure follows, as shown in Algorithm 1.

Using the ShDE for the RSKPCA algorithm (denoted as ShKPCA) allows for the restatement of the upper bounds presented in §3.5 in closed form, as long as the conditions

$$(k(a,b) - k(c,d))^2 \leq C_{\mathcal{X}}^k(\|a-b\|^2 + \|b-d\|^2). \tag{3.6.3}$$

Algorithm 2 Shadow Set Selection Procedure

Input: $\mathcal{X} = \{x_i\}_{i=1}^n$, bandwidth σ, and $\ell \in \mathbb{R}_+$.

Procedure:
Set $\mathrm{C} = \emptyset$, $\mathcal{W} = \emptyset$, $m = 0$, and

$$\varepsilon = \sigma/\ell. \tag{3.6.2}$$

while $\mathcal{X} \neq \emptyset$ **do**
  Let c be first element of $\mathcal{X}$.
  Find shadow set $S = \{y \in \mathcal{X} \ : \ \|y - c\| < \varepsilon\}$.
  Update center set $\mathrm{C} = \mathrm{C} \cup \{c\}$.
  Update weight set $\mathcal{W} = \mathcal{W} \cup \{|S|\}$.
  Set $\mathcal{X} = \mathcal{X} \backslash S$.
end while

Output:
Center set $\mathrm{C} = \{c_1, \ldots, c_m\}$, and weight set $\mathcal{W} = \{w_1, \ldots, w_m\}$.

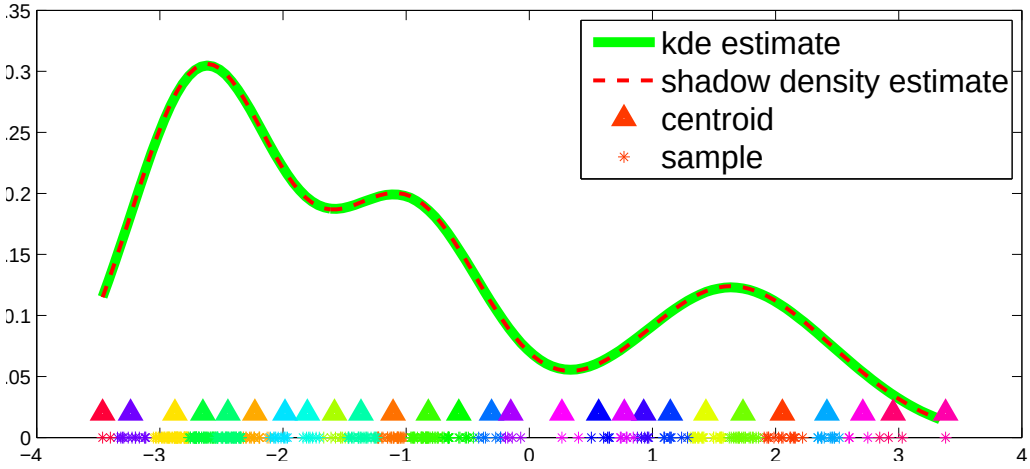


Figure 8: Visualization of the data, the shadow centers, and the associated KDE and ShKDE.

$$k(x, y) = \varphi\left(\frac{\|x - y\|^p}{\sigma^p}\right), \tag{3.6.4}$$

are met. The Laplacian and Gaussian, in particular, satisfy (3.6.4) for $\varphi(s) = e^{-s}$. The constant $C_{\mathcal{X}}^k$ is $\frac{1}{\sigma^2}$ for the Laplacian, and is $\frac{1}{2\sigma^2}$ for the Gaussian [115].

Proposition 3.6.1. ***(MMD Worst Case Bound 2)*** *Let the RSDE be computed by the ShDE procedure, n be the number of samples, $\mathcal{X}$ be defined as above, $\bar{\mathcal{C}}$ be the*

quantized dataset, k satisfy (3.6.4). Then

$$\mathrm{MMD}(\mathcal{X},\ \bar{\mathcal{C}}\ )_b \leq \sqrt{2\left(\kappa - \varphi\left(\frac{1}{\ell^p}\right)\right)}. \tag{3.6.5}$$

Proof. Following the proof of Proposition 3.5.1, and using (3.6.4), we have that

$$\varphi\left(\frac{\|x_{\max} - c_j\|^p}{\sigma^p}\right) = \varphi\left(\frac{1}{\ell^p}\right),$$

and therefore

$$\mathrm{MMD}_{\mathrm{sub},j} \leq w_j\sqrt{2\left(\kappa - \varphi\left(\frac{1}{\ell^p}\right)\right)}.$$

Summing over all the subproblems proves the proposition. □

The ShDE+RSKPCA procedure creates a matrix $\widetilde{K}$ that acts as an $m \times m$ surrogate for the quantized kernel matrix $\overline{K}_{ij} = k(c_{\vartheta(i)}, c_{\vartheta(j)})$, for $i, j = 1 \ldots n$. Exploiting the quantization effect, the following proposition bounds the eigenvalue difference between the two spectral decompositions .

Proposition 3.6.2. *Let k be such that (3.6.4) holds, and let λ_i and $\bar{\lambda}_i$ be the eigenvalues of the normalized matrices K and $\overline{K}$ respectively. Then for a constant $C^k_{\mathcal{X}}$,*

$$\sum_{i=1}^{n}(\lambda_i - \bar{\lambda}_i)^2 \leq 2C^k_{\mathcal{X}}\left(\frac{\sigma}{\ell}\right)^2.$$

Proof. Let A and B be normal matrices, with eigenvalues λ_i and $\bar{\lambda}_i$ respectively. The Hoffman-Wielandt inequality states that

$$\sum_{i=1}^{n}(\lambda_i - \bar{\lambda}_i)^2 \leq \|A - B\|_F^2.$$

Let $A = K$ and $B = \overline{K}$. Using equation (3.6.3) element-wise, the Frobenius norm $\|A-B\|_F^2$ is upper bounded. Furthermore, (3.6.4), more specifically defines the upper bound given that K and $\overline{K}$ are normalized. □

Similar to the previous section, we can derive an upper bound on the Hilbert-Schmidt error between the two operators $\widehat{C}_{\mathcal{H}}$ and $\widetilde{C}_{\mathcal{H}}$.

Theorem 3.6.1. ***(Hilbert-Schmidt Worst Case Bound 2)*** *Let the RSDE be computed by the ShDE procedure, and let $\widehat{C}_{\mathcal{H}}$ and $\widetilde{C}_{\mathcal{H}}$ be defined as above. Then*

$$\left\|\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}} \leq 2\sqrt{\kappa}\sqrt{2\left(\kappa - \varphi\left(\frac{1}{\ell^p}\right)\right)}. \tag{3.6.6}$$

Proof. Following the proof of Theorem 3.5.1, we get the upper bound

$$\left\|\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}} \leq \frac{1}{n}\sum_{j=1}^{m}\sum_{x_i \in S_j} 2\sqrt{\kappa}\|\epsilon_i\|_{\mathcal{H}}. \tag{3.6.7}$$

Let ϵ' be the maximizer of the error. From the proof in Proposition 3.6.1,

$$\|\epsilon'\|_{\mathcal{H}} = \sqrt{2\left(\kappa - \varphi\left(\frac{1}{\ell^p}\right)\right)}.$$

Summing over the terms in (3.6.7) proves the theorem. □

Theorem 3.6.1 shows that the centroid error in $\mathcal{H}$ is the key to the performance of the learning algorithm, and that the error is controlled solely in terms of the parameter ℓ. The independence of the performance from the weights shows that ShDE effectively learns the percentage of the data that needs to be retained based on the value of ℓ, which is dependent on the kernel and not the data. Finally, ℓ controls both the MMD and operator approximations, implying that the density estimate used in the shadow density procedure is sensible for learning in the eigenspace. Using this result, the following proposition follows.

Proposition 3.6.3. *Let $\widehat{C}_{\mathcal{H}}$ and $\widetilde{C}_{\mathcal{H}}$ be symmetric, positive (finite) Hilbert-Schmidt operators on $\mathcal{H}$ defined as above, and assume that K_n has simple nonzero eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_n$. Let $D > 0$ be an integer such that $\lambda_D > 0$, $\delta_D = \frac{1}{2}(\lambda_D - \lambda_{D+1})$. If $2\sqrt{\kappa}\|\epsilon'\|_{\mathcal{H}} < \delta_D/2$, then*

$$\|P^D(K_n) - P^D(\overline{K}_n)\|_{\text{HS}} \leq \frac{2\sqrt{2\kappa\left(\kappa - \varphi\left(\frac{1}{\ell^p}\right)\right)}}{\delta_D}, \tag{3.6.8}$$

where $P^D(A)$ denotes the projection onto the D-dimensional eigenspace of $A \in \text{HS}(\mathcal{H})$, associated to the largest eigenvalues.

Proof. Follows from Theorem 3 in [119] and Theorem 3.6.1. □

3.7 On the Use of Operator Bounds in the Nonasymptotic Setting

It's clear that the bounds (3.5.5) and (3.6.6) are nonasymptotic, while the convergence of the operator $\widehat{C}_{\mathcal{H}}$ to $C_{\mathcal{H}}$ is only shown in §3.2 in an asymptotic sense. For practitioners interested in strict (probabilistic) guarantees using Algorithm 1 with respect to the ideal operator $C_{\mathcal{H}}$, the following result for nonseparable, complex Hilbert spaces $\mathcal{H}$ can be employed.

Theorem 3.7.1. *(**De Vito et al** [107]) Let $\mathcal{X}$ be sampled i.i.d. from ν, and let $\widehat{C}_{\mathcal{H}}$ and $C_{\mathcal{H}}$ be defined as above. Then, with confidence $1 - 2e^{-\tau}$,*

$$\left\| C_{\mathcal{H}} - \widehat{C}_{\mathcal{H}} \right\|_{\text{HS}} \leq \frac{2\sqrt{2}\kappa\sqrt{\tau}}{\sqrt{n}}. \tag{3.7.1}$$

Corollary 3.7.1. *Let the conditions for the above theorem hold, and let $\widetilde{C}_{\mathcal{H}}$ be defined as above. Then,*

$$\left\| C_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} \right\|_{\text{HS}} \leq \frac{2\sqrt{2}\kappa\sqrt{\tau}}{\sqrt{n}} + \frac{2\sqrt{\kappa}}{n} \sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(x_{\max_j}, c_j)\right)}, \tag{3.7.2}$$

where $x_{\max_j} = \arg\min_{x_i \in S_j} k(x_i, c_j)$, and S_j represents the set of points assigned to center c_j by Algorithm 1. If Algorithm 2 is used to generate the reduced set, the following bound holds:

$$\left\| C_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} \right\|_{\text{HS}} \leq \frac{2\sqrt{2}\kappa\sqrt{\tau}}{\sqrt{n}} + 2\sqrt{\kappa}\sqrt{2\left(\kappa - \varphi\left(\frac{1}{\ell^p}\right)\right)}, \tag{3.7.3}$$

where ℓ is the shadow density parameter.

Proof. Follows from an application of the triangle inequality and Theorem 3.5.1. □

The bound (3.7.2) shows that the distance in Hilbert-Schmidt norm between the ideal covariance operator $C_{\mathcal{H}}$ and the RsKPCA covariance operator $\widetilde{C}_{\mathcal{H}}$ can be made arbitrarily small. In addition, for guaranteed performance without having to compute an upper bound from the data, Algorithm 2 can be used.

3.8 Experimental Results

This section demonstrates the effectiveness of RSKPCA on real-world data. Approximation accuracy tests include eigenembedding and classification tasks with the Gaussian kernel. The datasets used and the bandwidths chosen are given in Table 2.

Table 2: Datasets used.

	german	**pendigits**	**usps**	**yale**
n	1,000	3,500	9,298	5,768
dim	24	16	256	520
classes	2	10	10	10
k	5	5	15	10
σ	30	120	18	17

Table 3: Training cost and storage comparison.

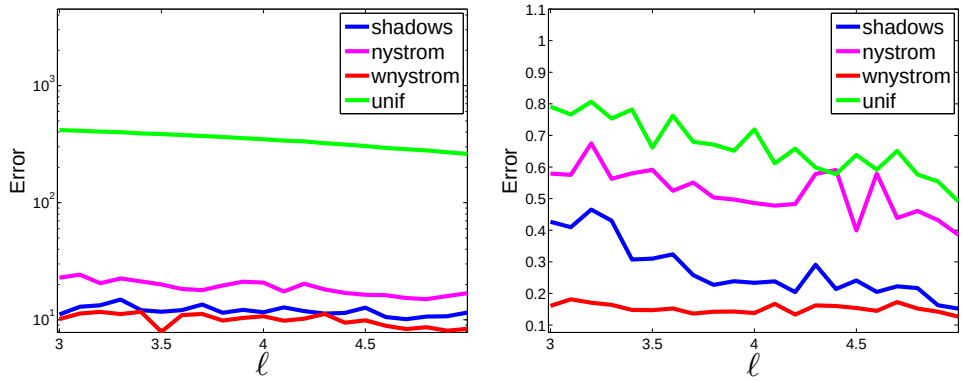
	ShDE+RSKPCA	Nyström	WNyström
time	$\mathcal{O}(mn + m^3)$	$\mathcal{O}(mn + m^3)$	$\mathcal{O}(mn + m^3)$
space	$\mathcal{O}(mr)$	$\mathcal{O}(nr)$	$\mathcal{O}(nr)$

Eigenembedding comparison with Nyström methods. This experiment demonstrates the fidelity of the eigenfunctions computed by ShDE+RSKPCA to those generated by KPCA. The capacity of generalization of the approximate eigenfunctions is tested. Using KPCA as the baseline, ShDE+RSKPCA is compared with three other methods: 1) subsampled KPCA, with bases chosen via uniform sampling, 2) the regular Nyström method, with bases chosen via uniform sampling, and 3) the

density-weighted Nyström (WNyström) method [116]. The experimental methodology is as follows. First, the KPCA model is trained on the entire dataset. Then, shadow, uniform, Nyström, and WNyström KPCA models are trained using 80% of the data for $\ell \in [3.0, 5.0]$, in increments of 0.1. The KPCA eigenfunction embedding is computed for the remaining 20% of the data for all the models, with rank $r = 5$. The embeddings are aligned with each other using the transform $\text{argmin}_{A \in \mathbb{R}^{r \times r}} \|O - \tilde{O}A\|_F$, where O is the matrix representing the KPCA embedding, and $\tilde{O}$ represents the approximate KPCA embedding. The Frobenius-norm difference of the embeddings and eigenvalues, the training and testing speedup, and the amount of data retained, are averaged over 50 runs for each ℓ, and are shown in Figures 9 and 10 for the **german** and **pendigits** datasets. As expected, while subsampled KPCA is faster in the training stage, it performs worse than any other method, implying that an appropriate weighting is necessary to approximate the eigenfunctions of KPCA. For larger values of ℓ, ShDE+RSKPCA always performs well when it comes to approximating the eigenvalues and eigenfunctions of the operator. In terms of eigenembedding accuracy, using ANOVA with a value of $\alpha = 0.05$, ShDE+RSKPCA is better than the Nyström embeddings after $\ell = 3.2(3.3)$ and no worse than the WNyström embeddings after $\ell = 4.0(4.8)$ for **pendigits** (**german**), and asymptotically approaches the KPCA baseline. While slower than the Nyström method for training, ShDE+RSKPCA is faster than KPCA for training, and achieves significant testing speedups. It does so by retaining a subset of the data via selection of ℓ. Computational savings scale with dataset size and redundancy. The savings are exploited in the next experimental comparison.

KPCA classification comparison with Nyström methods. This experiment examines the effectiveness of ShDE+RSKPCA for classification compared with the Nyström methods used previously. Classification utilizes the k-nn classifier with $k = 3$, using 10-fold cross-validation. The accuracy, training and testing speedups,

and the percentage of data retained, are reported. The results are shown in Figures 11 and 12 for the **usps** and **yale** methods respectively (none = KPCA). For the k-nn classification case, ShDE+RSKPCA has competitive accuracy with the Nyström methods, while providing significant training and testing speedups. The training speedup over the Nyström method in this case is because the eigenembedding of the data needs to be computed as part of the k-nn classifier training. As can be seen, through ShDE+RSKPCA, significant reductions in both training and evaluation time are achieved with minimal performance loss for large, redundant datasets. Competetive overall speedups and performace were achieved versus Nyström methods.

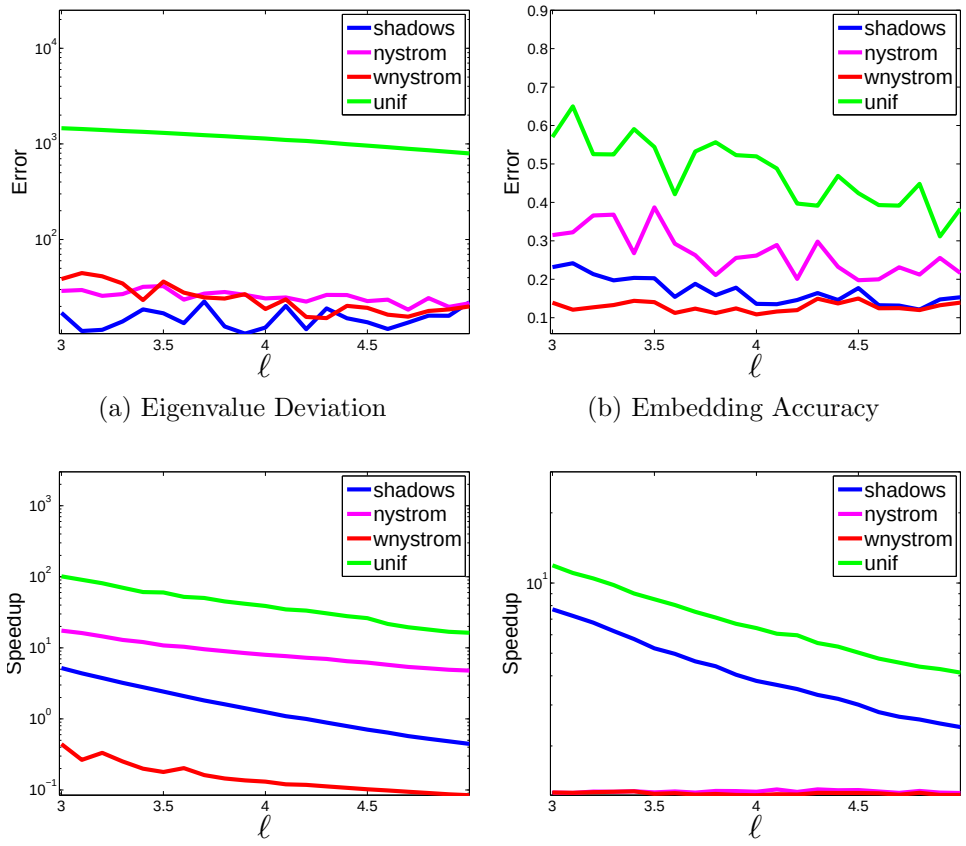


(a) Eigenvalue Deviation (b) Embedding Accuracy

(c) Eigen speedup (d) Testing speedup

Figure 9: Eigenembedding comparison w/Nyström methods for **german** as ℓ is varied ($n_t = 800$).

Note that the data retained here, Fig. 13(c,d), is less than 10% for $\ell \in [3, 5]$,



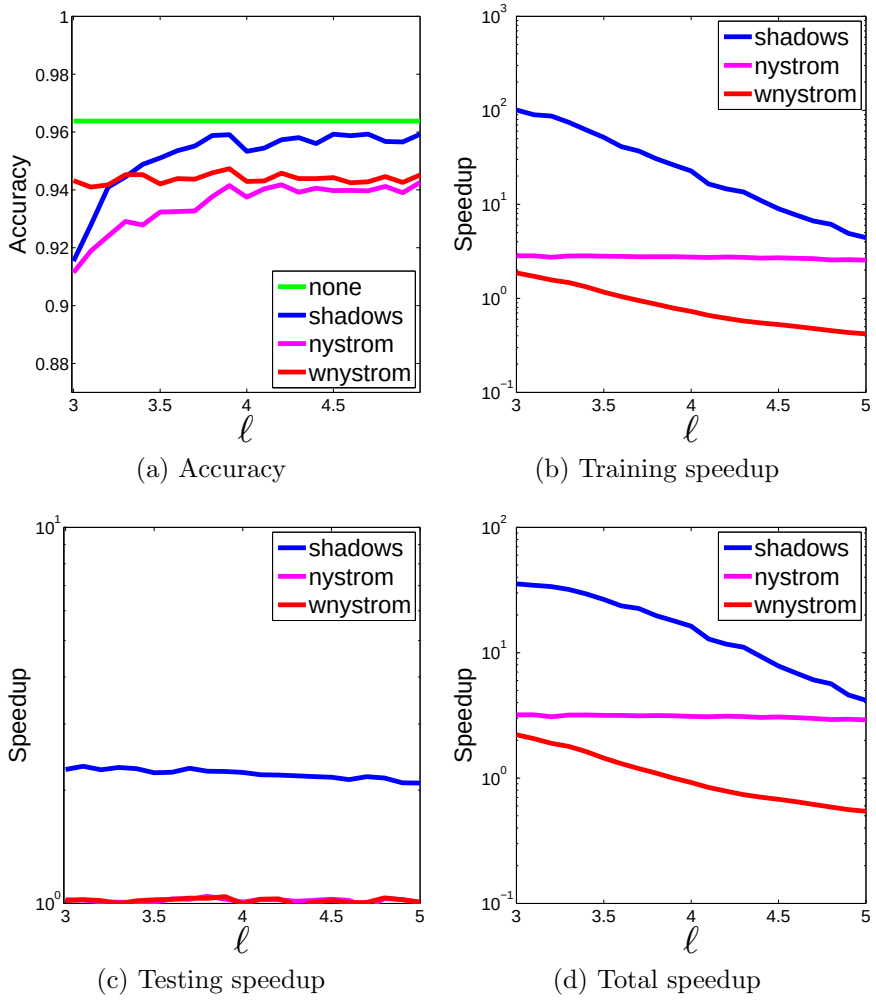
(a) Eigenvalue Deviation (b) Embedding Accuracy

(c) Eigen speedup (d) Testing speedup

Figure 10: Eigenembedding comparison w/Nyström for **pendigits** as ℓ is varied ($n_t = 2,800$).

implying noticeable speedup in the KPCA step of the classifier (during training and evaluation).

RSKPCA with different RSDE schemes. RSKPCA is performed using alternative RSDEs to demonstrate the influence of the RSDE algorithm on accuracy, Figs. 14 and 15. Following [116], k-means provides a means to generate an RSDE at a time complexity of $\mathcal{O}(mn)$ (but tends to be slow due being iterative). Second, KDE paring [31] subsamples from the original dataset and computes the estimate from the reduced set, at an $\mathcal{O}(m)$ cost. Third, kernel herding is examined [15], which provides a mechanism to sample from a KDE using a nonlinear dynamical system. The samples are shown to be good representative samples. Their generation is $\mathcal{O}(n^2m)$. All



(a) Accuracy (b) Training speedup

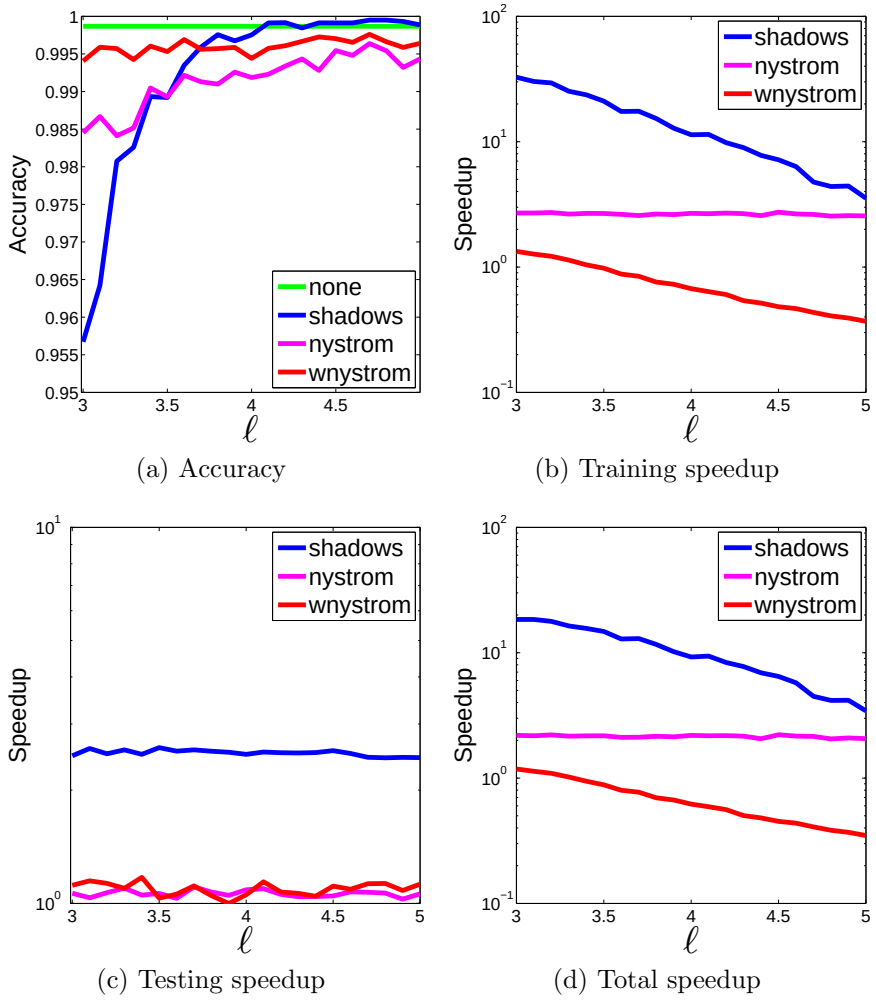
(c) Testing speedup (d) Total speedup

Figure 11: Classification comparison w/Nyström for **usps** as ℓ is varied ($n_t = 8,368$).

of these algorithms require the user to provide the number m. It can be seen that the quality of the RSDE does influence the accuracy for small ℓ, less so for larger ℓ. The center selection schemes that lead to improved accuracy are costlier than ShDE, thus decreasing training gains. Evaluation speedup is the same for all methods.

3.9 Conclusion

This chapter presented (1) a reduced-set KPCA algorithm for speeding up KPCA given a reduce set density estimate of the training data, and (2) a simple, efficient, single-pass algorithm for generating a suitable RSDE, called the shadow density estimate (ShDE), which relies on a user-selected parameter ℓ. The spectral decomposition

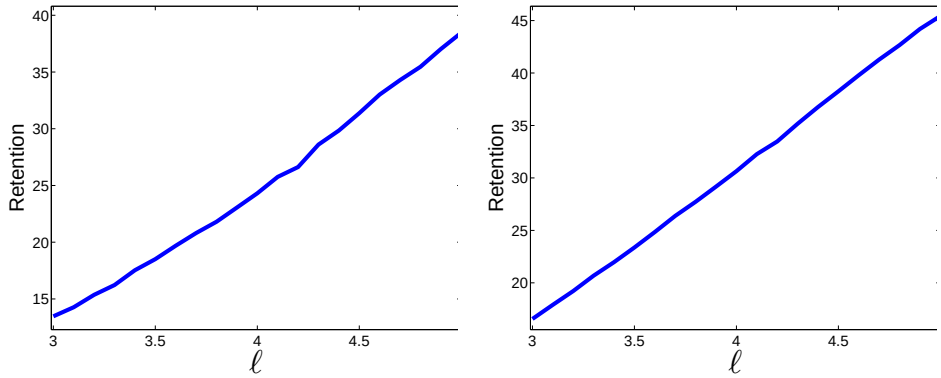


(a) Accuracy (b) Training speedup

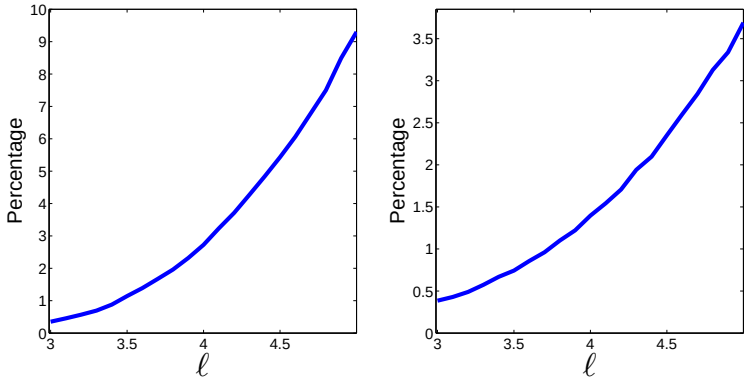
(c) Testing speedup (d) Total speedup

Figure 12: Classification comparison w/Nyström methods for **yale** as ℓ is varied ($n_t = 5,191$).

error was shown to be bounded and directly related to the bound of the empirical error of the ShDE. Through ShDE+RSKPCA, significant reductions in both training and evaluation time are achieved with minimal performance loss for large, redundant datasets. Competetive overall speedups and performance were achieved versus Nyström methods.

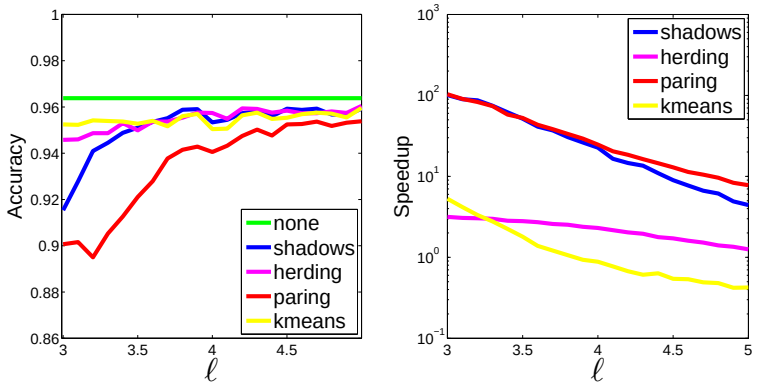


(a) **german** (b) **pendigits**



(c) **usps** (d) **yale**

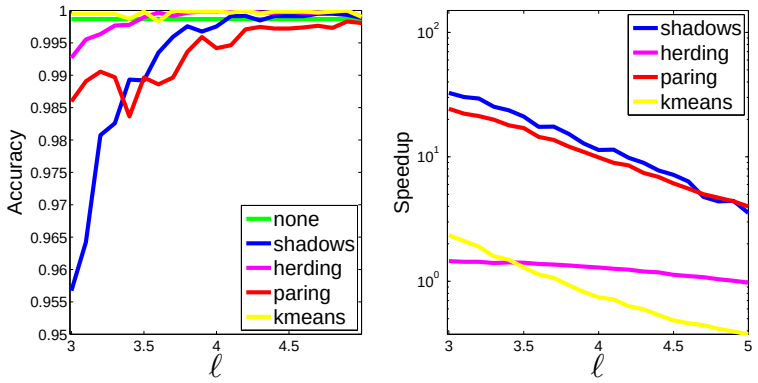
Figure 13: Percentage of data retained.



(a) Accuracy (b) Training speedup

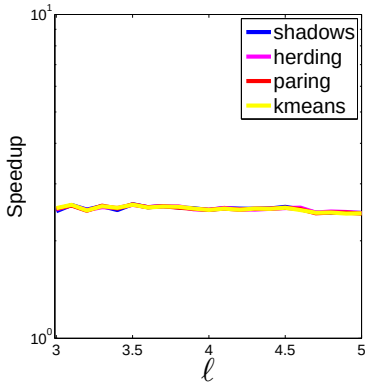
(c) Testing speedup (d) Total speedup

Figure 14: Classification comparisons w/RSDEs for **usps** as ℓ is varied ($n_t = 8,368$).

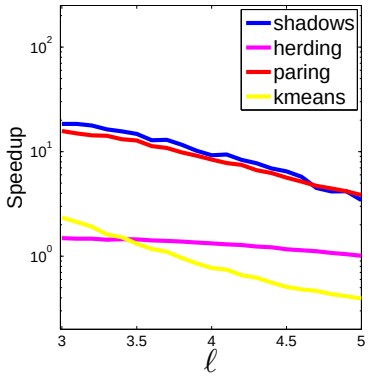


(a) Accuracy

(b) Training speedup



(c) Testing speedup



(d) Total speedup

Figure 15: Classification comparisons w/RSDEs for **yale** as ℓ is varied ($n_t = 5,191$).

CHAPTER IV

REDUCED-SET KERNEL PCA: APPLICATIONS

This chapter outlines some applications of RsKPCA for speeding up other kernel-based learning algorithms. In particular, we consider Gaussian process regression, diffusion maps, and kernel embeddings.

4.1 Gaussian Process Regression

A powerful, nonparametric method for regression uses the properties of Gaussian processes, which are stochastic processes $\{X_t; t \in T\}$, where each finite, linear combination of random variables is normally distributed. Gaussian processes are completely specified by their second-order statistics, which are defined by the so-called covariance kernel function. Since positive-definite kernel functions induce RKHSs, Gaussian-process regression (GPR) can be considered as an example of a kernel method. In this section, we give a description of GPR from a Bayesian point of view [78], and formulate a reduced-set approximation for GPR, which we denote as RSGPR. We also compute bounds on the deviation of RSGPR from GPR in a couple of metrics.

4.1.1 Formulation and Reduced-Set Approximation

GPR is an extension of Bayesian linear regression. Given a dataset $\mathcal{Z} = \{(x_i, y_i)\}_1^n$, where $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$, consider the problem of recovering a function $f(x)$ from observations y such that the following holds

$$f(x) = w^T x \quad y = f(x) + \epsilon, \tag{4.1.1}$$

where $w \in \mathbb{R}^d$ and $\epsilon \in \mathcal{N}(0, \omega^2)$. The noise assumption in conjunction with the linear model (4.1.1) gives rise to a likelihood function for the data, given by $p(y|\mathcal{Z}, w) =$

$\Pi_i p(y_i|x_i, w) = \mathcal{N}(X^T w, \omega^2 I)$. Here, X is the $d \times n$ data matrix. If a prior is put on the weights, $w \sim \mathcal{N}(0, \Sigma_w)$, the solution for the posterior $p(y|X)$ can be computed in closed form; in particular, the posterior distribution for the weights is obtained, and the predictive distribution $p(f_*|x_*, X, y)$ can be computed, which is also normally distributed. The inference procedure just described holds when the data is mapped to a higher-dimensional space using a fixed basis function network $\widehat{\psi}(x)$ of dimensionality N; in particular, the function is given by $f(x) = w^T \widehat{\psi}(x)$, and the predictive distribution is given by

$$f_*|x_*, X, y \sim \mathcal{N}\left(\frac{1}{\omega^2}\widehat{\psi}_*^T A^{-1}\widehat{\Psi} y, \widehat{\psi}_*^T A^{-1}\widehat{\psi}_*\right), \tag{4.1.2}$$

where $\widehat{\psi}_* := \widehat{\psi}(x_*)$, $\widehat{\Psi} := \widehat{\psi}(X)$, and $A := \omega^2 \widehat{\Psi}\widehat{\Psi}^T + \Sigma_p^{-1}$. If the dimensionality N of the network $\widehat{\psi}(x)$ is very high, the inversion of the matrix A becomes problematic. In this case, the dual formulation of the problem can be used, yielding the equation

$$f_*|x_*, X, y \sim \mathcal{N}\left(\widehat{\psi}_*^T \Sigma_p \widehat{\Psi}(K + \omega^2 I)^{-1} y, \widehat{\psi}_*^T \Sigma_p \widehat{\psi}_* - \widehat{\psi}_*^T \Sigma_p \widehat{\Psi}(K + \omega^2 I)^{-1} \widehat{\Psi}^T \Sigma_p \widehat{\psi}_*\right), \tag{4.1.3}$$

where $K := \widehat{\Psi}^T \Sigma_p \widehat{\Psi}$. Since the feature space always enters the computations in the form of inner products, the basis functions can be replaced by a covariance kernel $k(x, x') := \widehat{\psi}(x)^T \Sigma_p \widehat{\psi}(x)$. In particular, given a positive-definite kernel function $k(x, x')$, (4.1.3) holds, and can be interpreted as performing Bayesian linear regression in the feature space $\mathcal{H}$.

Note that the main bottleneck in the computation of (4.1.3) is the inversion of the regularized kernel matrix $K + \omega^2 I$. We can use the results in §3 to devise a reduced-set approximation to this kernel matrix, and have an algorithm that runs in $\mathcal{O}(nm + m^3)$ time, rather than $\mathcal{O}(n^3)$ time.

Algorithm 3 Reduced-Set Weighted Kernel Matrix Regularized Inverse

Input: Initial data $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, noise parameter ω^2.

Procedure:

1) Compute RSDE on $\mathcal{Z}$; yields center set $\bar{\mathcal{Z}} = \{\bar{z}_1, \ldots, \bar{z}_m\}$ and weights $W = \{w_1, \ldots, w_m\}$.
2) Compute $\bar{\mathcal{Z}} = \mathrm{C} \times \mathrm{D}$, where $\mathrm{C} = \{c_1, \ldots, c_m\}$, and $\mathrm{D} = \{d_1, \ldots, d_m\}$.
3) Create diagonal matrix $W = \mathrm{diag}(w_1, \ldots, w_m)$.
4) Compute weighted kernel matrix

$$\widetilde{K} \in \mathbb{R}^{m \times m}, \quad \widetilde{K} := W^{\frac{1}{2}} K^{\mathrm{C}} W^{\frac{1}{2}}$$

where $K^{\mathrm{C}}_{ij} := k(c_i, c_j)$.
5) Perform eigenvector decomposition $\widetilde{K} = U \Lambda U^T$
6) Reweight eigenvectors $\widetilde{U} = W^{1/2} U$.

Output:
Return regularized inverse

$$\widetilde{U}(\Lambda + \omega^2 I)^{-1} \widetilde{U}^T,$$

and $W = \{w_1, \ldots, w_m\}$, $\mathrm{C} = \{c_1, \ldots, c_m\}$, and $\mathrm{D} = \{d_1, \ldots, d_m\}$.

4.2 Reduced-Set Approximation Bounds

Like kernel PCA, the main object of study in GPR is the covariance operator (3.2.6), i.e.

$$\widehat{C}_{\mathcal{H}} := \frac{1}{n} \sum_{i=1}^{n} \psi(x_i) \otimes \psi(x_i),$$

which is represented in dual form by the kernel matrix K. Similarly, the quantized covariance operator (3.3.4), i.e.

$$\widetilde{C}_{\mathcal{H}} = \frac{1}{n} \sum_{i=1}^{n} \psi(c_{\vartheta(i)}) \otimes \psi(c_{\vartheta(i)}),$$

where $\vartheta : \{1, \ldots, n\} \to \{1, \ldots, m\}$ is the data-to-center mapping induced by the reduced-set approximation algorithm, is represented in dual form by the quantized kernel matrix $\overline{K}_{ij} = k(c_{\vartheta(i)}, c_{\vartheta(i)})$, for $i, j = 1 \ldots n$.

We now compute bounds on the deviation between GPR and RSGPR. First, we have the following result.

Algorithm 4 Reduced-Set Gaussian Process Regression

Input: Initial data $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, noise parameter ω^2.

Procedure:

1) Apply Algorithm 3 to get $\left(\widetilde{K} + \omega^2 I\right)^{-1}$, $W = \{w_1, \ldots, w_m\}$, C $= \{c_1, \ldots, c_m\}$, and D $= \{d_1, \ldots, d_m\}$.

2) Compute coefficients

$$\alpha = \widetilde{U}(\Lambda + \omega^2 I)^{-1}\widetilde{U}^T d,$$

where $d \in \mathbb{R}^m$ is a column vector representation of D.

Output:

Compute approximate GP mean as

$$\widetilde{f}(x) := \sum_{i=1}^{m} \alpha_i k(c_i, x), \tag{4.1.4}$$

and approximate posterior variance as

$$\widetilde{\nu}(x) := k(x, x) - k_x^T \widetilde{U}(\Lambda + \omega^2 I)^{-1}\widetilde{U}^T k_x, \tag{4.1.5}$$

where $k_x := [k(c_1, x), \ldots, k(c_m, x)]^T$.

Proposition 4.2.1. *Let $\widehat{C}_{\mathcal{H}}$ and $\widetilde{C}_{\mathcal{H}}$ be defined as in (3.2.6) and (3.3.4), let $f \in \mathcal{H}$, and let the map ϑ be available using an arbitrary RSDE. Then*

$$\left\langle \psi(x), \left(\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right) f \right\rangle_{\mathcal{H}} \leq \frac{\kappa}{n} \|f\|_{\mathcal{H}} \sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(x_{max_j}, c_j)\right)}. \tag{4.2.1}$$

If the RSDE is computed using Algorithm 2, then

$$\left\langle \psi(x), \left(\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right) f \right\rangle_{\mathcal{H}} \leq 2\kappa^2 \|f\|_{\mathcal{H}} \sqrt{2\left(\kappa - \varphi\left(\frac{1}{\ell^p}\right)\right)}. \tag{4.2.2}$$

Proof. Let $\|A\|_{\text{op}}$ denote the operator norm for operators $A : \mathcal{H} \to \mathcal{H}$. Then via the properties of norms,

$$\begin{aligned}
\left\langle \psi(x), \left(\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right) f \right\rangle_{\mathcal{H}} &\leq \|\psi(x)\|_{\mathcal{H}} \left\| \left(\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right) f \right\|_{\mathcal{H}} \\
&\leq \|\psi(x)\|_{\mathcal{H}} \|f\|_{\mathcal{H}} \left\| \widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} \right\|_{\text{op}} \\
&\leq \|\psi(x)\|_{\mathcal{H}} \|f\|_{\mathcal{H}} \left\| \widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} \right\|_{\text{HS}}.
\end{aligned}$$

Using Theorems 3.5.1 and 3.6.1 proves the result. □

We can also prove a result that bounds the difference in the actual *hypotheses* computed by the algorithms. Here, we define the hypotheses associated to the two covariance operators above as $h(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$ and $\widetilde{h}(x) = \sum_{i=1}^{n} \widetilde{\alpha}_i k(c_{\vartheta(i)}, x)$. Assume further that the kernel is normalized, such that

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} k(x, y) dx dy = C,$$

and for finite sets,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j) \leq C.$$

For the sake of simplicity, we normalize $k(x, y)$ explicitly by dividing by n in our calculations.

Theorem 4.2.1. *Let h and $\widetilde{h}$ be defined as above, $\sup_{x' \in \Omega} k(x', x) \leq \kappa$, and $\|y\|_\infty \leq M$. Then*

$$\left| h(x) - \widetilde{h}(x) \right| \leq \frac{\kappa M}{n\omega^4} \left(\sqrt{\boldsymbol{\Upsilon}^2 + 4\kappa^2 n \boldsymbol{\Upsilon}} + \sqrt{n}\omega^2 \boldsymbol{\Upsilon} \right), \tag{4.2.3}$$

where

$$\boldsymbol{\Upsilon} := \sum_{j=1}^{m} w_j \sqrt{2 \left(\kappa - k(x'_j, c_j) \right)}, \tag{4.2.4}$$

$k(x'_j, c_j)$ represents the smallest value that can occur over all possible values of x_j in the center sets S_i, and ω^2 represents the variance of the i.i.d. noise in the observations y_i.

Proof. We extend the proof of a similar theorem in [103]. Let $K_{ij} := n^{-2} k(x_i, x_j)$ and $\widetilde{K}_{ij} := n^{-2} k(c_{\vartheta(i)}, c_{\vartheta(j)})$; then, we have

$$\begin{aligned} \widetilde{\alpha} - \alpha &= (n^{-2}\widetilde{K} + \omega^2 I)^{-1} y - (n^{-2} K + \omega^2 I)^{-1} y \\ &= -\left[(n^{-2}\widetilde{K} + \omega^2 I)^{-1} (n^{-2}\widetilde{K} - n^{-2} K)(n^{-2} K + \omega^2 I)^{-1} \right] y. \end{aligned}$$

Therefore, we can bound

$$\begin{aligned}\|\widetilde{\alpha}-\alpha\| &\leq \left\|\frac{1}{n^2}\widetilde{K}+\omega^2 I\right\|_2 \left\|\frac{1}{n^2}\left(\widetilde{K}-K\right)\right\|_2 \left\|\frac{1}{n^2}K+\omega^2 I\right\|_2 \|y\| \\ &\leq \left\|\frac{1}{n^2}\widetilde{K}+\omega^2 I\right\|_2 \left\|\frac{1}{n^2}\left(\widetilde{K}-K\right)\right\|_F \left\|\frac{1}{n^2}K+\omega^2 I\right\|_2 \|y\| \\ &\leq \frac{\|\widetilde{K}-K\|_F\|y\|}{n\lambda_{\min}\left(\frac{1}{n^2}\widetilde{K}+\omega^2 I\right)\lambda_{\min}\left(\frac{1}{n^2}K+\omega^2 I\right)},\end{aligned}$$

where $\lambda_{\min}(A)$ is the smallest singular value of the matrix A. Now, we can bound these singular values as

$$\begin{aligned}\lambda_{\min}\left(\frac{1}{n^2}K+\omega^2 I\right) &= \lambda_{\min}\left(\frac{1}{n^2}K+\frac{n^2}{n^2}\omega^2 I\right) \\ &= \frac{1}{n^2}\lambda_{\min}\left(K+n^2\omega^2 I\right) \\ &\geq \frac{1}{n^2}\left(n^2\omega^2\right) \\ &\geq \omega^2.\end{aligned}$$

Plugging this estimate back above, we get

$$\|\widetilde{\alpha}-\alpha\| \leq \frac{\|\widetilde{K}-K\|_F\|y\|}{n^2\omega^4},$$

The hypothesis h derived with the exact kernel matrix can be written as $h(x) = \sum_{i=1}^{n}\alpha_i k(x_i,x) = \alpha^T k_x$, where k_x is a vector representation of the kernel evaluation, while $\widetilde{h}$ can be written as $\widetilde{h}(x) = \sum_{i=1}^{n}\widetilde{\alpha}_i k(c_{\vartheta(i)},x) = \alpha^T\widetilde{k}_x$, where $\widetilde{k}_x$ is a vector representation of the equivalent kernel evaluation. We can write

$$\begin{aligned}\left|h(x)-\widetilde{h}(x)\right| &= \left\|\alpha^T k_x-\widetilde{\alpha}^T\widetilde{k}_x\right\| \\ &= \left\|\alpha^T k_x-\widetilde{\alpha}^T k_x+\widetilde{\alpha}^T k_x-\widetilde{\alpha}^T\widetilde{k}_x\right\| \\ &\leq \left\|\alpha^T k_x-\widetilde{\alpha}^T k_x\right\|+\left\|\widetilde{\alpha}^T k_x-\widetilde{\alpha}^T\widetilde{k}_x\right\| \\ &\leq \|\alpha-\widetilde{\alpha}\|\,\|k_x\|+\|\widetilde{\alpha}\|\|k_x-\widetilde{k}_x\|.\end{aligned}$$

The hypothesis h derived with the exact kernel matrix can be written as $h(x) = \frac{1}{n}\sum_{i=1}^{n}\alpha_i k(x_i,x) = \alpha^T k_x$, where k_x is a vector representation of the kernel evaluation,

while $\widetilde{h}$ can be written as $\widetilde{h}(x) = \frac{1}{n}\sum_{i=1}^{n} \widetilde{\alpha}_i k(c_{\vartheta(i)}, x) = \alpha^T \widetilde{k}_x$, where $\widetilde{k}_x$ is a vector representation of the equivalent kernel evaluation. Note that here, unlike when we first introduced the hypothesis, we have normalized the kernel evaluations. We can now write

$$\begin{aligned}
\left|h(x) - \widetilde{h}(x)\right| &= \left\|\alpha^T k_x - \widetilde{\alpha}^T \widetilde{k}_x\right\| \\
&= \left\|\alpha^T k_x - \widetilde{\alpha}^T k_x + \widetilde{\alpha}^T k_x - \widetilde{\alpha}^T \widetilde{k}_x\right\| \\
&\leq \left\|\alpha^T k_x - \widetilde{\alpha}^T k_x\right\| + \left\|\widetilde{\alpha}^T k_x - \widetilde{\alpha}^T \widetilde{k}_x\right\| \\
&\leq \|\alpha - \widetilde{\alpha}\| \, \|k_x\| + \|\widetilde{\alpha}\| \|k_x - \widetilde{k}_x\|.
\end{aligned}$$

Our goal is to bound these terms separately. Since κ is the maximum of the kernel, $\|k_x\| \leq \frac{\kappa}{\sqrt{n}}$, and $\|y\| \leq \sqrt{n}M$, the first term can be bounded as

$$\begin{aligned}
\|\alpha - \widetilde{\alpha}\| \, \|k_x\| &\leq \frac{\kappa}{\sqrt{n}} \|\alpha - \widetilde{\alpha}\| \\
&\leq \frac{\kappa \|y\| \|\widetilde{K} - K\|_F}{n\sqrt{n}\omega^4} \\
&\leq \frac{\kappa M \|\widetilde{K} - K\|_F}{n\omega^4}.
\end{aligned}$$

To bound the second, we have that

$$\begin{aligned}
\|\widetilde{\alpha}\| &\leq \left\| \left(\frac{1}{n^2} \widetilde{K} + \omega I \right)^{-1} \right\| \|y\| \\
&\leq \frac{\|y\|}{\lambda_{\min}\left(\frac{1}{n^2} \widetilde{K} + \omega^2 I \right)} \\
&\leq \frac{\sqrt{n} M}{\omega^2}.
\end{aligned}$$

Furthermore, note that

$$\begin{aligned}
\left\|k_x - \widetilde{k}_x\right\| &= \frac{1}{n}\left\|\begin{pmatrix} k(x_1,x) - k(c_{\vartheta(1)},x) \\ k(x_2,x) - k(c_{\vartheta(2)},x) \\ \vdots \\ k(x_n,x) - k(c_{\vartheta(n)},x)\end{pmatrix}\right\| \\
&= \frac{1}{n}\left\|\begin{pmatrix} \left\langle \psi(x_1) - \psi(c_{\vartheta(1)}), \psi(x)\right\rangle_{\mathcal{H}} \\ \left\langle \psi(x_2) - \psi(c_{\vartheta(2)}), \psi(x)\right\rangle_{\mathcal{H}} \\ \vdots \\ \left\langle \psi(x_n) - \psi(c_{\vartheta(n)}), \psi(x)\right\rangle_{\mathcal{H}}\end{pmatrix}\right\| \\
&= \frac{\sqrt{n}}{n}\max_i \left|\left\langle \psi(x_i) - \psi(c_{\vartheta(i)}), \psi(x)\right\rangle_{\mathcal{H}}\right| \\
&\leq \frac{1}{\sqrt{n}}\sum_{i=1}^{n} \|\psi(x)\|_{\mathcal{H}} \left\|\psi(x_i) - \psi(c_{\vartheta(i)})\right\|_{\mathcal{H}} \\
&\leq \frac{\kappa}{\sqrt{n}}\sum_{i=1}^{m} w_i \sqrt{2\left(\kappa - k(x_{\max_i}, c_i)\right)},
\end{aligned}$$

where $k(x_{\max_i}, c_i)$ represents the smallest value that can occur over all $x_j \in S_i$. Putting these terms together, we have that

$$\begin{aligned}
\left|h(x) - \widetilde{h}(x)\right| &\leq \frac{\kappa M \|\widetilde{K} - K\|_F}{n\omega^4} + \frac{\kappa M \sum_{i=1}^{m} w_i \sqrt{2\left(\kappa - k(x_{\max_i}, c_i)\right)}}{\sqrt{n}\omega^2} \\
&\leq \frac{\kappa M}{n\omega^4}\left(\|\widetilde{K} - K\|_F + \sqrt{n}\omega^2 \sum_{i=1}^{m} w_i \sqrt{2\left(\kappa - k(x_{\max_i}, c_i)\right)}\right) \\
&\leq \frac{\kappa M}{n\omega^4}\left(\|\widetilde{K} - K\|_F + \sqrt{n}\omega^2 \boldsymbol{\Upsilon}\right).
\end{aligned}$$

To connect the matrix norm above to our earlier work with Hilbert-Schmidt norms, first note that

$$\begin{aligned}
\frac{1}{n^2}\|\widetilde{K} - K\|_F^2 &= \frac{1}{n^2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(k(x_i,x_j) - k(c_{\vartheta(i)}, c_{\vartheta(j)})\right)^2\right) \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} k(x_i,x_j)^2 + k(c_{\vartheta(i)}, c_{\vartheta(j)})^2 - 2k(x_i,x_j)k(c_{\vartheta(i)}, c_{\vartheta(j)})\right).
\end{aligned}$$

Now, for operators from $\mathcal{H}$ to $\mathcal{H}$, the following holds:

$$\langle v \otimes u, b \otimes a \rangle_{\mathrm{HS}} = \langle u, a \rangle_{\mathcal{H}} \langle b, v \rangle_{\mathcal{H}}.$$

Therefore, the norm of the covariance operator can be computed as

$$\begin{aligned}
\left\| \widehat{C}_{\mathcal{H}} \right\|_{\mathrm{HS}}^2 &= \left\langle \frac{1}{n} \sum_{i=1}^{n} \psi(x_i) \otimes \psi(x_i), \frac{1}{n} \sum_{j=1}^{n} \psi(x_j) \otimes \psi(x_j) \right\rangle_{\mathrm{HS}} \\
&= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \psi(x_i) \otimes \psi(x_i), \psi(x_j) \otimes \psi(x_j) \rangle_{\mathrm{HS}} \\
&= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \psi(x_i), \psi(x_j) \rangle_{\mathcal{H}} \langle \psi(x_i), \psi(x_j) \rangle_{\mathcal{H}} \\
&= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j)^2.
\end{aligned}$$

Using the above computation, we have

$$\begin{aligned}
\left\| \widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} \right\|_{\mathrm{HS}}^2 &= \left\langle \widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}, \widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} \right\rangle_{\mathrm{HS}} \\
&= \left\langle \widehat{C}_{\mathcal{H}}, \widehat{C}_{\mathcal{H}} \right\rangle_{\mathrm{HS}} + \left\langle \widetilde{C}_{\mathcal{H}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\mathrm{HS}} - \left\langle \widehat{C}_{\mathcal{H}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\mathrm{HS}} - \left\langle \widetilde{C}_{\mathcal{H}}, \widehat{C}_{\mathcal{H}} \right\rangle_{\mathrm{HS}} \\
&= \frac{1}{n^2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left(k(x_i, x_j)^2 + k(c_{\vartheta(i)}, c_{\vartheta(j)})^2 - 2k(x_i, c_{\vartheta(j)}) k(c_{\vartheta(i)}, x_j) \right) \right).
\end{aligned}$$

This is similar to the Frobenius norm, except for the terms of the form

$$k(x_i, c_{\vartheta(j)}) k(c_{\vartheta(i)}, x_j),$$

which differ from

$$k(x_i, x_j) k(c_{\vartheta(i)}, c_{\vartheta(j)}). \tag{4.2.5}$$

The term (4.2.5) can be generated from the Hilbert-Schmidt norm of the rank-1 operator

$$\left\langle \psi(x_i) \otimes \psi(c_{\vartheta(i)}), \psi(x_j) \otimes \psi(c_{\vartheta(j)}) \right\rangle_{\mathrm{HS}}.$$

Therefore, we can define a new operator

$$\widehat{C}_{\mathcal{X}\mathrm{C}} := \frac{1}{n} \sum_{i=1}^{n} \psi(x_i) \otimes \psi(c_{\vartheta(i)}).$$

Then it can be shown that

$$\frac{1}{n^2}\|\widetilde{K} - K\|_F^2 = \left\|\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}}^2 + 2\left(\left\langle \widehat{C}_{\mathcal{H}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\text{HS}} - \left\langle \widehat{C}_{\mathcal{X}\text{C}}, \widehat{C}_{\mathcal{X}\text{C}} \right\rangle_{\text{HS}}\right).$$

To bound the RHS term, we have

$$\begin{aligned}
\left\langle \widehat{C}_{\mathcal{H}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\text{HS}} - \left\langle \widehat{C}_{\mathcal{X}\text{C}}, \widehat{C}_{\mathcal{X}\text{C}} \right\rangle_{\text{HS}} &= \left\langle \widehat{C}_{\mathcal{H}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\text{HS}} - \left\langle \widehat{C}_{\mathcal{X}\text{C}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\text{HS}} \\
&\quad + \left\langle \widehat{C}_{\mathcal{X}\text{C}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\text{HS}} - \left\langle \widehat{C}_{\mathcal{X}\text{C}}, \widehat{C}_{\mathcal{X}\text{C}} \right\rangle_{\text{HS}} \\
&= \left\langle \widehat{C}_{\mathcal{H}} - \widehat{C}_{\mathcal{X}\text{C}}, \widetilde{C}_{\mathcal{H}} \right\rangle_{\text{HS}} + \left\langle \widehat{C}_{\mathcal{X}\text{C}}, \widetilde{C}_{\mathcal{H}} - \widehat{C}_{\mathcal{X}\text{C}} \right\rangle_{\text{HS}} \\
&\leq \left\|\widehat{C}_{\mathcal{H}} - \widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}} \left\|\widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}} \\
&\quad + \left\|\widetilde{C}_{\mathcal{H}} - \widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}} \left\|\widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}}.
\end{aligned}$$

Now these terms need to be bounded as well. First, we have that

$$\begin{aligned}
\left\|\widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}} &= \frac{1}{n}\|\widetilde{K}\|_F \\
&\leq \kappa,
\end{aligned}$$

and

$$\left\|\widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}} \leq \kappa.$$

Therefore,

$$\begin{aligned}
\left\|\widehat{C}_{\mathcal{H}} - \widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}} &= \left\|\frac{1}{n}\sum_{i=1}^{n} \psi(x_i) \otimes (\psi(x_i) - \psi(c_{\vartheta(i)}))\right\|_{\text{HS}} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \left\|\psi(x_i) \otimes (\psi(x_i) - \psi(c_{\vartheta(i)}))\right\|_{\text{HS}} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \left\|\psi(x_i)\right\|_{\mathcal{H}} \left\|\psi(x_i) - \psi(c_{\vartheta(i)})\right\|_{\mathcal{H}} \\
&\leq \frac{\kappa}{n}\sum_{i=1}^{n} \sqrt{\kappa - k(x_i, c_{\vartheta(i)})} \\
&\leq \frac{\kappa}{n}\sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(x_{\max_j}, c_j)\right)}.
\end{aligned}$$

This leads to the bound

$$\begin{aligned}\frac{1}{n^2}\|\widetilde{K}-K\|_F^2 &\leq \left\|\widehat{C}_{\mathcal{H}}-\widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}}^2+2\left\|\widehat{C}_{\mathcal{H}}-\widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}}\left\|\widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}}\\ &\quad+2\left\|\widetilde{C}_{\mathcal{H}}-\widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}}\left\|\widehat{C}_{\mathcal{X}\text{C}}\right\|_{\text{HS}}\\ &\leq \left\|\widehat{C}_{\mathcal{H}}-\widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}}^2+\frac{4\kappa^2}{n}\sum_{j=1}^{m}w_j\sqrt{2\left(\kappa-k(x_{\max_j},c_j)\right)}.\end{aligned}$$

Multiplying both sides by n^2, using the result from Theorem 3.5.1, and taking square roots on both sides leads to

$$\begin{aligned}\|\widetilde{K}-K\|_F &\leq n\sqrt{\left\|\widehat{C}_{\mathcal{H}}-\widetilde{C}_{\mathcal{H}}\right\|_{\text{HS}}^2+\frac{4\kappa^2}{n}\sum_{j=1}^{m}w_j\sqrt{2\left(\kappa-k(x_{\max_j},c_j)\right)}}\\ &\leq n\sqrt{\left(\frac{1}{n}\sum_{j=1}^{m}w_j\sqrt{2\left(\kappa-k(x'_j,c_j)\right)}\right)^2+\frac{4\kappa^2}{n}\sum_{j=1}^{m}w_j\sqrt{2\left(\kappa-k(x'_j,c_j)\right)}}\\ &\leq \sqrt{\left(\sum_{j=1}^{m}w_j\sqrt{2\left(\kappa-k(x'_j,c_j)\right)}\right)^2+4\kappa^2 n\sum_{j=1}^{m}w_j\sqrt{2\left(\kappa-k(x'_j,c_j)\right)}}\\ &\leq \sqrt{\boldsymbol{\Upsilon}^2+4\kappa^2 n\boldsymbol{\Upsilon}},\end{aligned}$$

which proves the theorem. □

Writing the bound in Theorem 4.2.1 in asymptotic notation as $\mathcal{O}(\boldsymbol{\Upsilon}/\sqrt{n})$, we see that it depends primarily on the center approximation error $\boldsymbol{\Upsilon}$, and decreases as the number of samples grows.

4.3 Diffusion Maps

As mentioned in §3, modern datasets are characterized by big datasets in extremely high dimensional spaces. A simplifying assumption that is often made is that the data are actually embedded in a submanifold of the input space [55]. The subject of manifold learning is concerned with the design of algorithms to recover this manifold. One of the most common strategies for manifold learning involve graph-based algorithms, where the graphs are constructed by assigning links between two points

based on some notion of affinity [21]. Weighted graphs can be used to represent a notion of geometry based on the local similarity of the data. When combined with Markov chain techniques, random walks on graphs can be used to find structure in very complex geometries [22]. A subclass of these methods are the so-called kernel eigenmap methods, such as locally linear embedding [82], Laplacian eigenmaps [5] and Hessian eigenmaps [26], which can be thought of as performing kernel PCA on specially designed kernel matrices [35]. The central idea in these algorithms is that the eigenvectors of Markov transition matrices can be thought of as a new set of coordinates in Euclidean space, which leads to the graph to be embedded as a cloud of points in Euclidean space.

Kernel eigenmaps can be shown to constitute special cases of a general framework based on diffusion processes; the diffusion map algorithm is the implementation of this framework in practice [22]. In diffusion maps, the eigenfunctions of various Markov matrices are used to define a random walk on the data, in order to obtain new descriptions of datasets via a family of mappings. Different geometric representations of the data can be obtained by iterating Markov matrix of transition, or equivalently, by running the random walk forward. These maps relate the spectral properties of the diffusion process to the geometry of the data [65].

In this section, the diffusion map algorithm is outlined in the continuous realm. Then, the empirical version of the algorithm is derived using empirical measure approximations of the underlying probability density. Finally, a reduced-set approximation is derived, and the error in the MMD and the Hilbert-Schmidt norm between this reduced-set approximation and the empirical map is bounded.

4.3.1 Random Walks on Data

In the diffusion map algorithm, the goal is to define a random walk on the data. Let $(\Omega, \mathcal{A}, \nu)$ be a probability space, and let $\mathcal{X} \subset \Omega$ be the dataset sampled from ν. Let

$k_\zeta : \Omega \times \Omega \to \mathbb{R}$ be a symmetric, positive-definite kernel that represents a notion of similarity between points in $\mathcal{X}$. If the points in $\mathcal{X}$ represent the nodes in a graph, then k_ζ specifies a weight function on the graph, and thus a prior definition of the local geometry of $\mathcal{X}$ [22]. Using the graph defined by $(\mathcal{X}, k_\zeta)$, a reversible Markov chain on $\mathcal{X}$ can be constructed as follows. Let

$$p_\zeta(x) = \int_\Omega k_\zeta(x, y) d\nu(y). \tag{4.3.1}$$

be the local measure of the volume in the graph, and define a new kernel

$$M_f(x, y) = \frac{k_\zeta(x, y)}{p_\zeta(y)}. \tag{4.3.2}$$

This is a non-symmetric kernel with the conservation property

$$\int_\Omega M_f(x, y) d\nu(y) = 1. \tag{4.3.3}$$

Thus (4.3.2) can be viewed as the (forward) transition kernel of a Markov chain on $\mathcal{X}$, and induces a Chapman-Kolmogorov operator on functions $u \in \mathcal{L}^2(\Omega, \nu)$, which can be computed as

$$T_f[u](x) = \int_\Omega M_f(x, y) u(y) d\nu(y). \tag{4.3.4}$$

A symmetric version of (4.3.2) can be defined as

$$M_s(x, y) = \frac{k_\zeta(x, y)}{\sqrt{p_\zeta(x) p_\zeta(y)}}, \tag{4.3.5}$$

which induces another Chapman-Kolmogorov operator as

$$T_s[u](x) = \int_\Omega M_s(x, y) \phi(y) p(y) dy. \tag{4.3.6}$$

A backwards Chapman-Kolmogorov operator can also be defined as

$$T_b[v](x) = \int_\Omega M_f(y|x) v(y) p(y) dy. \tag{4.3.7}$$

If $u(x)$ is the probability of finding the system at location x at time $t = 0$, then $T_f[u]$ is the evolution of the probability to time $t = \zeta$, whereas if $v(y)$ is some function on

the space then $T_b[v](x)$ is the average value of the function at time ζ for a random walk that started at x. These operators obey the property

$$\langle T_f u, v\rangle_p = \langle u, T_b v\rangle_p, \tag{4.3.8}$$

while T_s is self-adjoint. The eigenfunctions of these operators are used to define the diffusion embedding. Since T_s is obtained via conjugation of the kernel M_s with $\sqrt{p_\zeta}$, its eigenfunctions can be scaled appropriately to recover those of T_s. For example, if $T_s\phi_s = \lambda\phi_s$, then the corresponding eigenfunctions for T_f are $\phi_f = \sqrt{p_\zeta}\phi_s$.

4.3.2 Continuous Diffusion Maps

Let the eigenexpansion of the kernel $M_f(x, y)$ be given by

$$M_s(x, y) = \sum_{\iota\geq 0} \lambda_\iota \phi_\iota(x)\phi_\iota(y),$$

where $\{\phi_\iota\}_{\iota\geq 0}$ is an orthobasis of $\mathcal{L}^2(\Omega, \nu)$. Then the basis of $M_f(x, y)$ is given by

$$M_f(x, y) = \sum_{\iota\geq 0} \lambda_\iota \psi_\iota(x)\varphi_\iota(y),$$

where $\psi_\iota(x) = \phi_\iota(x)/p_\zeta(x)$ and $\varphi_\iota(y) = \phi_\iota(x)p_\zeta(x)$. The iterations of the chain are then given by

$$M_f^t(x, y) = \sum_{\iota\geq 0} \lambda_\iota^t \psi_\iota(x)\varphi_\iota(y), \tag{4.3.9}$$

where t is the time parameter. The family of *diffusion distances* $\{D_t\}_{t\in\mathbb{N}}$ is then given by

$$D_t(x, y)^2 := \|M_f^t(x|\cdot) - M_f^t(y|\cdot)\|^2.$$

The diffusion distance reflects the connectivity of the data at a given scale. Using the eigendecomposition (4.3.9), it can be computed via

$$D_t(x, y) = \left(\sum_{\iota\geq 1} \lambda_\iota^{2t}(\psi_\iota(x) - \psi_\iota(y))^2\right)^{\frac{1}{2}}. \tag{4.3.10}$$

This allows for the creation of the *diffusion maps* $\{\Psi_t\}_{t\in\mathbb{N}}$ via

$$\Psi_t(x) := \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \cdots \\ \lambda_k^t \psi_k(x) \end{pmatrix}, \tag{4.3.11}$$

where the rank k is chosen beforehand. Here, the first eigenfunction, denoted by $\phi_0(x)$, is ignored.

4.3.3 Empirical Diffusion Maps

This section shows how to create the Gram matrices for the eigendecomposition of the empirical version of the self-adjoint operator (4.3.6), using the ideas presented in §3.3. First, we make note of the following lemma.

Lemma 4.3.1. *(**Lafon** [54]) Let $T_s : \mathcal{L}^2(\Omega,\nu) \to \mathcal{L}^2(\Omega,\nu)$ be defined as in (4.3.6). Then T_s is symmetric and positive semidefinite.*

This lemma implies that the kernel M_s induces a feature map to an RKHS $\mathcal{H}_\zeta$; in particular, if $\psi : \Omega \to \mathcal{H}$ is the feature map associated to the kernel k_ζ, then

$$\chi(x) := \frac{\psi(x)}{\sqrt{p_\zeta(x)}} \tag{4.3.12}$$

is the feature map associated to the kernel M_s. Given a dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$, consider the empirical measure

$$p(x) \approx \nu_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}. \tag{4.3.13}$$

Then the empirical version of (4.3.1) is given by

$$\widehat{p}_\zeta(x) = \frac{1}{n}\sum_{i=1}^{n} k_\zeta(x_i, x),$$

and the continuous version of the symmetric kernel (4.3.5) is given by

$$T_s(x,y) = \frac{k_\zeta(x,y)}{\sqrt{\sum_{\iota=1}^{n} k_\zeta(x,x_\iota)\sum_{\iota=1}^{n} k_\zeta(x_\iota,y)}}.$$

Algorithm 5 Diffusion Map

Input: Dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$, eigenvector rank $k+1$, and bandwidth ζ.

Procedure:

1) Given dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$, and eigenvector rank $k+1$, compute normalized kernel matrix

$$A \in \mathbb{R}^{n \times n}, \quad A_{ij} := \frac{k_\zeta(x_i, x_j)}{(\sqrt{\sum_{\iota=1}^n k_\zeta(x_\iota, x_i) \sum_{\iota=1}^n k_\zeta(x_\iota, x_j)})}.$$

2) Perform eigenvector decomposition $A\phi_i = \lambda_i \phi_i$.

3) Reweight to get the eigenvectors

$$\psi_i(j) = \frac{\phi_i(j)}{\sqrt{\sum_{\iota=1}^n k_\zeta(x_\iota, x_j)}},$$

where $\phi_i(j)$ is the jth component of the ith eigenvector.

Output:

Compute the diffusion embedding as

$$\Psi_t(x_j) := \begin{pmatrix} \lambda_1^t \psi_1(j) \\ \lambda_2^t \psi_2(j) \\ \cdots \\ \lambda_k^t \psi_k(j) \end{pmatrix}. \tag{4.3.15}$$

Therefre the Gram matrix for the symmetric kernel is given by

$$T_s(x_i, x_j) = \frac{k_\zeta(x_i, x_j)}{\sqrt{\sum_{\iota=1}^n k_\zeta(x_i, x_\iota) \sum_{\iota=1}^n k_\zeta(x_\iota, x_j)}}, \tag{4.3.14}$$

which is exactly the manner in which the matrix is computed in practice [94]. Using (4.3.14) with (4.3.11) yields Algorithm 5.

Using the notation in §3.3 and (4.3.12), we can define an empirical covariance operator associated with the diffusion kernel as

$$\widehat{D}_{\mathcal{H}_\zeta}^{(n)} := \frac{1}{n} \sum_{i=1}^n \widehat{\chi}(x_i) \otimes \widehat{\chi}(x_i), \tag{4.3.16}$$

an empirical approximation of an ideal covariance operator $D_{\mathcal{H}_\zeta}$, where

$$\widehat{\chi}(x) := \frac{\psi(x)}{\sqrt{\sum_{\iota=1}^n k_\zeta(x, x_\iota)}}. \tag{4.3.17}$$

Deriving a reduced-set approximation to (4.3.17) requires more work than RSKPCA, due to the dividing factor in the diffusion kernel $M_s(x, y)$. Consider the weighted

empirical measure

$$\widehat{\nu}_n = \frac{1}{n}\sum_{i=1}^{m} w_i \delta_{c_i},$$

where the reduced-set centers be given by $\mathrm{C} = \{c_1, \ldots, c_m\}$, and the measure induced by the data-to-center mapping $\vartheta : \{1, \ldots, n\} \to \{1, \ldots, m\}$. First, following [112], the diffusion kernel is reweighted in the following manner

$$\widetilde{k}_\zeta(x, y) = p^{1/2}(x) k_\zeta(x, y) p^{1/2}(y) \tag{4.3.18}$$

$$\approx \widehat{\nu}_n^{1/2}(x) k_\zeta(x, y) \widehat{\nu}_n^{1/2}(y). \tag{4.3.19}$$

This kernel has the same eigenvalues as $k_\zeta(x, y)$, and will be used for the diffusion map. Note that the kernel smoothing estimate (4.3.1) for this new kernel is given by

$$\widetilde{p}_\zeta(x) = \int_\Omega \widetilde{k}_\zeta(x, y) dy. \tag{4.3.20}$$

The density $p(x)$ is implicit in the kernel $\widetilde{k}_\zeta(x, y)$. Given an RSDE, (4.3.20) becomes

$$\widetilde{p}_\zeta(x) = \frac{1}{n}\int \sum_{i=1}^{m}\sum_{j=1}^{m} w_i^{1/2} \delta(c_i - x) k_\zeta(x, y) w_j^{1/2} \delta(c_j - y) dy \tag{4.3.21}$$

$$= \frac{1}{n}\int \sum_{i=1}^{m}\sum_{j=1}^{m} w_i^{1/2} w_j^{1/2} \delta(c_i - x) k_\zeta(x, c_j). \tag{4.3.22}$$

To recover the sample at c_i, one simply integrates out over x. Following the same line of thought, the discretized version of the reweighted kernel is given by

$$\widetilde{k}_\zeta(c_i, c_j) = \frac{1}{n}\sqrt{w_i} k_\zeta(c_i, c_j) \sqrt{w_j}. \tag{4.3.23}$$

Note that in the equation

$$\frac{\widetilde{k}_\zeta(c_i, c_j)}{\sqrt{\widetilde{p}_\zeta(c_i)\widetilde{p}_\zeta(c_j)}},$$

the weights cancel. Putting these steps together yields Algorithm 6. The covariance operator associated with the reduced-set approximation is given by

$$\widetilde{D}_{\mathcal{H}_\zeta}^{(n)} := \frac{1}{n}\sum_{i=1}^{m} w_i \widetilde{\chi}(c_i) \otimes \widetilde{\chi}(c_i), \tag{4.3.24}$$

Algorithm 6 Reduced-Set Diffusion Map

Input: Dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$, rank $k+1$, bandwidth ζ, and no. of centers m.

Procedure:
1) Compute RSDE $\mathrm{C} = \{c_1, \ldots, c_m\}$ and $w = \{w_1, \ldots, w_m\}$.
2) Compute normalized weighted kernel matrix

$$\widetilde{A} \in \mathbb{R}^{m \times m}, \quad \widetilde{A}_{ij} := \frac{k(c_i, c_j)}{(\sqrt{\sum_{\iota=1}^{n} \sqrt{w_\iota} k(c_\iota, c_i) \sum_{\iota=1}^{n} \sqrt{w_\iota} k(x_\iota, c_j)})}.$$

3) Perform eigenvector decomposition $\widetilde{A}\tilde{\phi}_i = \lambda_i \tilde{\phi}_i$.
4) Reweight to get the eigenvectors $\widehat{\phi}_i = W^{-1/2}\tilde{\phi}_i$.
5) Compute

$$\widehat{\psi}_i(j) = \frac{\widehat{\phi}_i(j)}{\sqrt{\sum_{\iota=1}^{m} w_\iota k(c_\iota, c_j)}},$$

where $\widehat{\phi}_i(j)$ is the jth component of the ith eigenvector.

Output:
Compute the diffusion embedding

$$\widehat{\Psi}_t(x_j) := \begin{pmatrix} \lambda_1^t \widehat{\psi}_1(j) \\ \lambda_2^t \widehat{\psi}_2(j) \\ \cdots \\ \lambda_k^t \widehat{\psi}_k(j) \end{pmatrix}. \tag{4.3.26}$$

where

$$\widetilde{\chi}(x) := \frac{\psi(x)}{\sqrt{\sum_{\iota=1}^{m} w_i k_\zeta(x, c_\iota)}}. \tag{4.3.25}$$

4.3.4 Out-of-Sample Extensions

The diffusion embedding defined by (4.3.11) is useful for studying the geometry of a given dataset. It is undefined in the case one needs to extrapolate the given map to out-of-sample points. Since the training is quite expensive, one can imagine training on a subset of the data and extrapolating the geometry to unseen points in order to perform the same analysis one could given more computational resources. A commonly used tool for this task is the Nyström method [4, 105].

For the training data $x_i \in \mathcal{X}$, the following holds

$$\sum_{x_i \in \mathcal{X}} \frac{k(x_i, x_j)}{w_i} \psi_\iota(x_i) = \lambda_\iota \psi_\iota(x_j).$$

Therefore the Nyström approximation to the eigenfunctions is given by

$$\widehat{\Psi}_t(x) := \begin{pmatrix} \lambda_1^t \widehat{\psi}_1(x) \\ \lambda_2^t \widehat{\psi}_2(x) \\ \cdots \\ \lambda_k^t \widehat{\psi}_k(x) \end{pmatrix}, \tag{4.3.27}$$

where

$$\widehat{\psi}_\iota(x) = \sum_{x_i \in \mathcal{X}} \frac{k(x_i, x)}{w_i} \psi_\iota(x_i).$$

4.3.5 Bounding Approximation Error

This section derives bounds on the MMD error and the Hilbert-Schmidt error between the averaging operators created by the original diffusion map, and the reduced-set diffusion map.

We use the notation

$$d_{x_i} := \frac{\psi(x_i)}{\sqrt{\left\langle \sum_{j=1}^n \psi(x_j), \psi(x_i) \right\rangle_{\mathcal{H}}}}, \tag{4.3.28}$$

and

$$d_{c_i} := \frac{\psi(c_i)}{\sqrt{\left\langle \sum_{j=1}^m w_j \psi(c_j), \psi(c_i) \right\rangle_{\mathcal{H}}}}. \tag{4.3.29}$$

Then the following theorems can be proven.

Theorem 4.3.1. *Let $\mathcal{H}_\zeta$ be the RKHS associated to M_s. Then*

$$\begin{aligned} \mathrm{MMD}_{\mathcal{H}_\zeta}(\mathcal{X}, \mathrm{C}) &:= \left\| \frac{1}{n} \sum_{i=1}^n (d_{x_i} - d_{c_i}) \right\|_{\mathcal{H}_\zeta} \\ &\leq \frac{1}{n} \sum_{j=1}^m \sum_{x_i \in S_j} \sqrt{\frac{\kappa \rho_i^- (\rho_i + \rho_i^+) - 2\rho_i^+}{\rho_i \rho_i^+ \rho_i^-}}, \end{aligned} \tag{4.3.30}$$

where S_j are the sets associated to the jth center, and where

$$\rho_i := C_\zeta \sqrt{\sum_{j=1}^{m} w_j k_\zeta(c_i, c_j)}, \tag{4.3.31}$$

$$\rho_i^+ := C_\zeta \sqrt{\sum_{j=1}^{m} w_j k_+(c_i, c_j)}, \tag{4.3.32}$$

$$\rho_i^- := C_\zeta \sqrt{\sum_{j=1}^{m} w_j k_-(c_i, c_j)}, \tag{4.3.33}$$

and

$$k_+(x, y) := \exp\left(\frac{-\|x-y\|^2}{4\zeta}\right) \exp\left(-\frac{\|c_i - c_j\|}{2\zeta}\right), \tag{4.3.34}$$

$$k_-(x, y) := \exp\left(\frac{-\|x-y\|^2}{4\zeta}\right) \left(\exp\left(-\frac{\|c_i - c_j\|}{2\zeta}\right)\right)^{-1}. \tag{4.3.35}$$

Proof. Define $e_i := d_{x_i} - d_{c_i}$ and

$$R_{ij} := \|c_i - c_j\|_2 \tag{4.3.36}$$

$$\wp := \max_{x_i \in \mathcal{X},\ c_j \in S_i} \|x_i - c_j\|_2. \tag{4.3.37}$$

We can write the profile for the diffusion kernel as

$$k_\zeta(x, y) = \varphi\left(\frac{\|x-y\|^2}{4\zeta}\right),$$

where

$$\varphi(x) := e^{-x}.$$

Then we have that

$$\begin{aligned} k_\zeta(x_i, c_j) &\leq \varphi\left(\frac{(R_{ij} - \wp)^2}{4\zeta}\right) \\ &\leq \varphi\left(\frac{R_{ij}^2 + \wp^2 - 2R_{ij}\wp}{4\zeta}\right) \\ &\leq \varphi\left(\frac{R_{ij}^2}{4\zeta}\right) \varphi\left(\frac{\wp^2}{4\zeta}\right) \varphi\left(-\frac{2R_{ij}\wp}{4\zeta}\right) \\ &\leq \frac{\varphi\left(\frac{R_{ij}^2}{4\zeta}\right) \varphi\left(\frac{\wp^2}{4\zeta}\right)}{\varphi\left(\frac{2R_{ij}}{4\zeta}\right)^{2\wp}}. \end{aligned}$$

Using the same strategy, we get

$$k_\zeta(x_i, c_j) \geq \varphi\left(\frac{R_{ij}^2}{4\zeta}\right) \varphi\left(\frac{\wp^2}{4\zeta}\right) \varphi\left(\frac{2R_{ij}}{4\zeta}\right)^{2\wp}.$$

Define new kernels and a constant

$$\check{k}(c_i, c_j) := \exp\left(-\frac{\|c_i - c_j\|}{2\zeta}\right) \tag{4.3.38}$$

$$k_-(c_i, c_j) := \frac{k_\zeta(c_i, c_j)}{\check{k}(c_i, c_j)} \tag{4.3.39}$$

$$k_+(c_i, c_j) := k_\zeta(c_i, c_j)\check{k}(c_i, c_j) \tag{4.3.40}$$

$$C_\zeta := \varphi\left(\frac{\wp^2}{4\zeta}\right). \tag{4.3.41}$$

Therefore, we get the bounds

$$C_\zeta^2 k_+(c_i, c_j) \leq k_\zeta(x_i, x_j) \leq C_\zeta^2 k_-(c_i, c_j) \tag{4.3.42}$$

$$C_\zeta k_+(c_i, c_j) \leq k_\zeta(x_i, c_j) \leq C_\zeta k_-(c_i, c_j). \tag{4.3.43}$$

We can write

$$\begin{aligned}\|e_i\|_{\mathcal{H}}^2 &= \langle d_{x_i} - d_{c_i}, d_{x_i} - d_{c_i}\rangle_{\mathcal{H}} \\ &= \left(\langle d_{x_i}, d_{x_i}\rangle_{\mathcal{H}} - \langle d_{x_i}, d_{c_i}\rangle_{\mathcal{H}}\right) + \left(\langle d_{c_i}, d_{c_i}\rangle_{\mathcal{H}} - \langle d_{x_i}, d_{c_i}\rangle_{\mathcal{H}}\right).\end{aligned}$$

We will bound these terms individually. The first one can be written as

$$\frac{\kappa}{\sum_{j=1}^n k(x_j, x_i)} - \frac{k(x_i, c_i)}{\sqrt{\sum_{\alpha=1}^n k(x_\alpha, x_i) \sum_{\beta=1}^n k(c_\beta, c_i)}}.$$

Using (4.3.42) and (4.3.43), we have

$$\begin{aligned}\left(\langle d_{x_i}, d_{x_i}\rangle_{\mathcal{H}} - \langle d_{x_i}, d_{c_i}\rangle_{\mathcal{H}}\right) &\leq \frac{\kappa}{\rho_i^+} - \frac{1}{\rho_i \rho_i^-} \\ &\leq \frac{\kappa\rho_i^-\rho_i - \rho_i^+}{\rho_i \rho_i^+ \rho_i^-},\end{aligned}$$

where ρ_i, ρ_i^+ and ρ_i^- are defined as in (4.3.31), (4.3.32) and (4.3.33) respectively. Similarly, it can be shown that

$$\left(\langle d_{c_i}, d_{c_i}\rangle_{\mathcal{H}} - \langle d_{x_i}, d_{c_i}\rangle_{\mathcal{H}}\right) \leq \frac{\kappa\rho_i^- - 1}{\rho_i^- \rho_i}.$$

Therefore,

$$\|e_i\|_{\mathcal{H}}^2 \leq \frac{\kappa\rho_i^-(\rho_i+\rho_i^+)-2\rho_i^+}{\rho_i\rho_i^+\rho_i^-}. \tag{4.3.44}$$

Taking square-roots on both sides and accumulating terms appropriately proves the theorem. □

Theorem 4.3.2. *Let the covariance operator $\widehat{D}_{\mathcal{H}_\zeta}^{(n)}$ associated to M_s be defined by (4.3.16), and the reduced-set covariance operator $\widetilde{D}_{\mathcal{H}_\zeta}^{(n)}$ be defined by (4.3.24). Then*

$$\left\|\widehat{D}_{\mathcal{H}_\zeta}^{(n)}-\widetilde{D}_{\mathcal{H}_\zeta}^{(n)}\right\|_{\mathrm{HS}} \leq \frac{1}{n}\sum_{j=1}^{m}\sum_{x_i\in S_j}\frac{\kappa(\rho_i+\rho_i^+)}{\rho_i\rho_i^+}\sqrt{\frac{\kappa\rho_i^-(\rho_i+\rho_i^+)-2\rho_i^+}{\rho_i\rho_i^+\rho_i^-}} \tag{4.3.45}$$

Proof. Using ideas from the proofs of Theorems 4.3.1 and 3.5.1, we have that

$$\left\|\widehat{D}_{\mathcal{H}_\zeta}^{(n)}-\widetilde{D}_{\mathcal{H}_\zeta}^{(n)}\right\|_{\mathrm{HS}} \leq \left\|\frac{1}{n}\sum_{i=1}^{n}\langle\cdot,d_{x_i}\rangle_{\mathcal{H}}e_i\right\|_{\mathrm{HS}}+\left\|\frac{1}{n}\sum_{i=1}^{n}\langle\cdot,e_i\rangle_{\mathcal{H}}d_{c_i}\right\|_{\mathrm{HS}}.$$

Thus

$$\begin{aligned}
\left\|\frac{1}{n}\sum_{i=1}^{n}\langle\cdot,d_{x_i}\rangle_{\mathcal{H}}e_i\right\|_{\mathrm{HS}} &\leq \frac{1}{n}\sum_{i=1}^{n}\left\|\frac{\psi(x_i)}{\sqrt{\left\langle\sum_{j=1}^{n}\psi(x_j),\psi(x_i)\right\rangle_{\mathcal{H}}}}\otimes e_i\right\|_{\mathrm{HS}}\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\left\|\frac{\psi(x_i)}{\sqrt{\left\langle\sum_{j=1}^{n}\psi(x_j),\psi(x_i)\right\rangle_{\mathcal{H}}}}\right\|_{\mathcal{H}}\|e_i\|_{\mathcal{H}}\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sum_{j=1}^{n}\langle\psi(x_j),\psi(x_i)\rangle_{\mathcal{H}}}\|\psi(x_i)\|_{\mathcal{H}}\|e_i\|_{\mathcal{H}}\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\frac{\kappa}{\sum_{j=1}^{n}k(x_j,x_i)}\|e_i\|_{\mathcal{H}}\\
&\leq \frac{1}{n}\sum_{i=1}^{n}\frac{\kappa}{\rho_i^+}\sqrt{\frac{\kappa\rho_i^-(\rho_i+\rho_i^+)-2\rho_i^+}{\rho_i\rho_i^+\rho_i^-}}.
\end{aligned}$$

The other term can be bounded similarly:

$$\begin{aligned}\left\|\frac{1}{n}\sum_{i=1}^{n}\langle\cdot,e_i\rangle_{\mathcal{H}}d_{c_i}\right\|_{\mathrm{HS}} &\leq \frac{1}{n}\sum_{i=1}^{n}\|d_{c_i}\|_{\mathcal{H}}\|e_i\|_{\mathcal{H}} \\ &\leq \frac{1}{n}\sum_{i=1}^{n}\frac{\kappa}{\sum_{j=1}^{m}k(c_j,c_i)}\|e_i\|_{\mathcal{H}} \\ &\leq \frac{1}{n}\sum_{i=1}^{n}\frac{\kappa}{\rho_i}\sqrt{\frac{\kappa\rho_i^{-}(\rho_i+\rho_i^{+})-2\rho_i^{+}}{\rho_i\rho_i^{+}\rho_i^{-}}}.\end{aligned}$$

Summing the terms together proves the theorem. □

Note that unlike the RSKPCA case, the upper bound (4.3.30) does not have the same form as (4.3.45).

4.4 Kernel Embeddings

Many modern applications of signal processing and machine learning deal with large volumes of continuous-valued data with complex statistical features such as multimodality and skewness. Modeling these features presents a problem for most existing frameworks such as graphical models, which often rely on parametric assumptions and/or the computation of linear relations in the variables, resulting in models that differ significantly from the generating processes. A recent development in the theory of kernel methods attempts to address this modeling deficiency by generalizing the embedding of measures to kernel embeddings of *conditional distributions* [98]. The key idea is to map conditional distributions to infinite-dimensional feature spaces using kernels; all subsequent comparisons of distributions can be achieved using feature space operations. These embeddings are motivated by the use of conditional distributions in probabilistic graphical models and filtering; algorithms for kernel belief propagation and kernel hidden Markov models are two examples of an application of conditional distribution embeddings [97, 99].

While these algorithms exhibit state-of-the-art performance on a variety of tasks, they suffer from $\mathcal{O}(n^3)$ training time and $\mathcal{O}(n^2)$ evaluation time. In this section, we

extend the ideas used in §3.2 to create an approximation algorithm for conditional distribution embeddings with a training time of $\mathcal{O}(nm + m^3)$ and an evaluation time of $\mathcal{O}(m^2)$.

4.4.1 Formulation

Recall that a probability distribution $p(x)$ can be represented as an element of the RKHS $\mathcal{H}$ via the mean map

$$\mu_X := \mathbb{E}\left[k(\cdot, X)\right] = \mathbb{E}\left[\psi(X)\right] = \int_\Omega \psi(X) dp(X), \tag{4.4.1}$$

where $\psi : \Omega \to \mathcal{H}$ is associated to the positive-definite kernel $k : \Omega \times \Omega \to \mathbb{R}$. The mean embedding of the distribution captures the mean of all functions $f \in \mathcal{H}$ with respect to the distribution $p(x)$ via the projection $\langle \mu_X, f \rangle_\mathcal{H} := \mathbb{E}_X\left[f(X)\right]\ \forall f \in \mathcal{H}$. To generalize the embedding to joint distributions, tensor products can be used similar to §3.2 to get

$$\mathcal{C}_{XY} := \int_{\Omega \times \Omega} \psi(x) \otimes \psi(y) dp(x, y), \tag{4.4.2}$$

where it's assumed for simplicity that the variables (x, y) share the same input space and kernel. Note that (4.4.2) is an element of the tensor product feature space $\mathcal{H} \otimes \mathcal{H}$; as before, we have that $\langle \psi(x) \otimes \psi(y), \psi(x') \otimes \psi(y') \rangle_{\mathcal{H}\otimes\mathcal{H}} = k(x, x')k(y, y')$. One immediate application of this joint distribution embedding is to check whether two random variables are independent, via the *Hilbert-Schmidt independence criterion*

$$\mathrm{HSIC}(X, Y) := \left\| \mathcal{C}_{XY} - \mu_X \otimes \mu_Y \right\|^2_{\mathcal{H}\otimes\mathcal{H}}. \tag{4.4.3}$$

It has been shown that (4.4.3) is zero if and only if the random variables X and Y are independent [96]. The kernel embedding of a conditional distribution $p(Y|X)$ is defined as

$$\mu_{Y|x} := \mathbb{E}_{Y|x}\left[\psi(Y)\right] = \int_\Omega \psi(y) p(y|x). \tag{4.4.4}$$

The conditional expectation of a function $f \in \mathcal{H}$ can be computed as $\mathbb{E}_{Y|x}[\psi(Y)] = \langle f, \mu_{Y|x} \rangle_{\mathcal{H}}$. Note that (4.4.4) is not a single element in $\mathcal{H}$, but varies according to the value of x. The embedding $\mu_{Y|x}$ is related to a specific operator via the equation

$$\mathcal{C}_{X|Y} := \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}, \tag{4.4.5}$$

which leads to

$$\mu_{Y|x} = \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1}\psi(x). \tag{4.4.6}$$

In practice, a regularized version of the inverse in (4.4.5) is computed, for numerical stability.

The empirical version of the conditional embedding operator, defined on sample sets $\mathcal{X}$ and $\mathcal{Y}$, sampled from the random variables X and Y respectively, is computed as

$$\widehat{\mathcal{C}}_{X|Y} = \Psi \left(K + \gamma I\right)^{-1} \Phi^T, \tag{4.4.7}$$

where $\Psi := (\psi(y_1), \ldots, \psi(y_n))$ and $\Phi := (\psi(x_1), \ldots, \psi(x_n))$ are feature matrices, $K = \Phi^T\Phi$ is the Gram matrix for the dataset $\mathcal{X}$, and γ is a regularization parameter. Equation (4.4.7) leads to

$$\widehat{\mu}_{Y|x} = \sum_{i=1}^{n} \beta_i(x)\psi(y_i)\boldsymbol{\beta}(x), \tag{4.4.8}$$

where

$$\boldsymbol{\beta}(x) = (\beta_1(x), \ldots, \beta_n(x))^T = \left(K + \gamma I\right)^{-1} k_x, \tag{4.4.9}$$

and $k_x = (k(x_1, x) \ldots, k(x_n, x))^T$. This conditional embedding operator can be used to derive a series of kernel rules for conditional probability, some of which will be approximated in the next section.

Algorithm 7 Reduced-Set Kernel Conditional Mean

Input: Initial data $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, noise parameter ω^2.

Procedure:

1) Apply Algorithm 3 to get $\left(\widetilde{K} + \omega^2 I\right)^{-1}$, $W = \{w_1, \ldots, w_m\}$, C $= \{c_1, \ldots, c_m\}$, and D $= \{d_1, \ldots, d_m\}$.

2) Compute coefficients

$$\boldsymbol{\beta}(x) := (\beta_1(x), \ldots, \beta_m(x))^T = \left(\widetilde{K} + \omega^2 I\right)^{-1} k_x, \tag{4.4.10}$$

where $k_x := (k(c_1, x) \ldots, k(c_m, x))^T$.

Output:

Compute conditional mean as

$$\widetilde{\mu}_{Y|x} := \sum_{i=1}^{m} \beta_i k(d_i, x). \tag{4.4.11}$$

4.4.2 Reduced-Set Approximations

The ideas developed in §4.1 can be used to compute an approximation algorithm for kernel conditional mean embedding, as shown in Algorithm 7. At the heart of this algorithm is the reduced-set conditional embedding operator; we now derive a bound on the deviation of this approximation from the empirical conditional embedding operator in a similar manner to the work we have done before. First, we prove the following extension of Theorem 3.5.1.

Proposition 4.4.1. *Let $\vartheta : \{1, \ldots, n\} \to \{1, \ldots, m\}$ be the data-to-center mapping generated by computing an RSDE on $\mathcal{X} \times \mathcal{Y}$. Define $\widehat{\mathcal{C}}_{XY}$ and $\widetilde{\mathcal{C}}_{XY}$ as*

$$\widehat{\mathcal{C}}_{XY} := \frac{1}{n} \sum_{i=1}^{n} \psi(x_i) \otimes \psi(y_i) \tag{4.4.12}$$

$$\widetilde{\mathcal{C}}_{XY} := \frac{1}{n} \sum_{i=1}^{n} \psi(c_{\vartheta(i)}) \otimes \psi(y_{\vartheta(i)}). \tag{4.4.13}$$

Then

$$\left\| \widehat{\mathcal{C}}_{XY} - \widetilde{\mathcal{C}}_{XY} \right\|_{\text{HS}} \leq \frac{1}{n} \left(\boldsymbol{\Upsilon} + \boldsymbol{\Upsilon}' \right), \tag{4.4.14}$$

where

$$\boldsymbol{\Upsilon} := \sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(x_{max_j}, c_j)\right)} \tag{4.4.15}$$

$$\boldsymbol{\Upsilon}' := \sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(y_{max_j}, d_j)\right)} \tag{4.4.16}$$

Proof. Since $k_{x_i}, k_{y_i} \in \mathcal{H}$, define residuals $\epsilon_i := k_{x_i} - k_{c_{\vartheta(i)}}$ and $\epsilon_i' := k_{y_i} - k_{d_{\vartheta(i)}}$. Then

$$\begin{aligned}
\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}} &= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{y_i}\rangle_{\mathcal{H}} k_{x_i} - \langle\cdot, k_{d_{\vartheta(i)}}\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{x_i}\rangle_{\mathcal{H}} k_{x_i} - \langle\cdot, k_{y_i} - \epsilon_i'\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\langle\cdot, k_{y_i}\rangle_{\mathcal{H}} \epsilon_i + \langle\cdot, \epsilon_i'\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right).
\end{aligned}$$

Using the properties of the norm,

$$\left\|\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right\|_{\mathrm{HS}} \leq \frac{1}{n}\sum_{i=1}^{n}\left\|\langle\cdot, k_{y_i}\rangle_{\mathcal{H}}\epsilon_i\right\|_{\mathrm{HS}} + \frac{1}{n}\sum_{i=1}^{n}\left\|\langle\cdot, \epsilon_i'\rangle_{\mathcal{H}} k_{c_{\vartheta(i)}}\right\|_{\mathrm{HS}}.$$

Using the results in the proof of Theorem 3.5.1, we have that

$$\left\|\widehat{C}_{\mathcal{H}} - \widetilde{C}_{\mathcal{H}}\right\|_{\mathrm{HS}} \leq \frac{1}{n}\sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(x_{\max_j}, c_j)\right)} + \frac{1}{n}\sum_{j=1}^{m} w_j \sqrt{2\left(\kappa - k(y_{\max_j}, d_j)\right)},$$

which proves the proposition. □

We now prove a bound on the deviation between the conditional embedding operator $\widehat{\mathcal{C}}_{X|Y}$ and its reduced-set approximation $\widetilde{\mathcal{C}}_{X|Y}$.

Theorem 4.4.1. *Let $\widehat{\mathcal{C}}_{X|Y}$, $\widetilde{\mathcal{C}}_{X|Y}$ be the the empirical conditional embedding operator and its reduced-set approximation respectively, associated to the sample sets $\mathcal{X}, \mathcal{Y}$, $\widehat{\mathcal{C}}_{YX}$, $\widetilde{\mathcal{C}}_{YX}$ be the cross-covariance operators associated to the sample sets $\mathcal{X}, \mathcal{Y}$, and let $\widehat{\mathcal{C}}_{XX}$, $\widetilde{\mathcal{C}}_{XX}$ be the covariance operators associated to the sample set $\mathcal{X}$, regularized by $\gamma \in \mathbb{R}_+$ (i.e. their eigenvalues are remapped as $\lambda_i \mapsto \lambda_i + \gamma$). Then*

$$\left\|\widehat{\mathcal{C}}_{X|Y} - \widetilde{\mathcal{C}}_{X|Y}\right\|_{\mathrm{op}} \leq \frac{1}{\gamma n}\left(\boldsymbol{\Upsilon} + \boldsymbol{\Upsilon}' + \frac{n}{\gamma}\sqrt{\boldsymbol{\Upsilon}^2 + 4\kappa^2 n \boldsymbol{\Upsilon}}\right),$$

where $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Upsilon}'$ are defined as in (4.4.15) and (4.4.16) respectively.

Proof. First, we have that

$$\begin{aligned}\left\|\widehat{\mu}_{Y|x} - \widetilde{\mu}_{Y|x}\right\|_{\text{op}} &= \left\|\widehat{\mathcal{C}}_{YX}\widehat{\mathcal{C}}_{XX}^{-1} - \widetilde{\mathcal{C}}_{YX}\widetilde{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} \\ &\leq \left\|\widehat{\mathcal{C}}_{YX}\widehat{\mathcal{C}}_{XX}^{-1} - \widetilde{\mathcal{C}}_{YX}\widehat{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} + \left\|\widetilde{\mathcal{C}}_{YX}\widehat{\mathcal{C}}_{XX}^{-1} - \widetilde{\mathcal{C}}_{YX}\widetilde{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} \\ &\leq \left\|\widehat{\mathcal{C}}_{YX} - \widetilde{\mathcal{C}}_{YX}\right\|_{\text{op}} \left\|\widehat{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} + \left\|\widehat{\mathcal{C}}_{XX}^{-1} - \widetilde{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} \left\|\widetilde{\mathcal{C}}_{YX}\right\|_{\text{op}}\end{aligned}$$

These four terms need to be bounded separately. First, since the operator norm, for self-adjoint, compact operators, corresponds to the largest eigenvalue, we have that

$$\begin{aligned}\left\|\widehat{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} &\leq \left\|\lambda_{\min}\left(\widehat{\mathcal{C}}_{XX}^{-1}\right)\right\|_{\text{op}} \\ &\leq \gamma^{-1}.\end{aligned}$$

Utilizing the same trick used in the proof of Theorem 4.2.1, we have that

$$\begin{aligned}\left\|\widehat{\mathcal{C}}_{XX}^{-1} - \widetilde{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} &= \left\|\widehat{\mathcal{C}}_{XX}^{-1}\left(\widehat{\mathcal{C}}_{XX} - \widetilde{\mathcal{C}}_{XX}\right)\widetilde{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} \\ &\leq \left\|\widehat{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} \left\|\widehat{\mathcal{C}}_{XX} - \widetilde{\mathcal{C}}_{XX}\right\|_{\text{op}} \left\|\widetilde{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} \\ &\leq \gamma^{-2}\left\|\widehat{\mathcal{C}}_{XX} - \widetilde{\mathcal{C}}_{XX}\right\|_{\text{op}} \\ &\leq \gamma^{-2}\left\|\widehat{\mathcal{C}}_{XX} - \widetilde{\mathcal{C}}_{XX}\right\|_{\text{HS}} \\ &\leq \frac{1}{\gamma^2 n}\sqrt{\boldsymbol{\Upsilon}^2 + 4\kappa^2 n\boldsymbol{\Upsilon}}.\end{aligned}$$

Since

$$\begin{aligned}\left\|\widetilde{\mathcal{C}}_{YX}\right\|_{\text{op}} &\leq \left\|\widetilde{\mathcal{C}}_{YX}\right\|_{\text{HS}} \\ &\leq \left\|\overline{K}\right\|_F \\ &\leq n,\end{aligned}$$

where $\overline{K}_{ij} := k(c_i, d_j)$, we have that

$$\left\|\widehat{\mathcal{C}}_{XX}^{-1} - \widetilde{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} \left\|\widetilde{\mathcal{C}}_{YX}\right\|_{\text{op}} \leq \frac{1}{\gamma^2}\sqrt{\boldsymbol{\Upsilon}^2 + 4\kappa^2 n\boldsymbol{\Upsilon}}.$$

Using the result from the previous proposition leads to the bound

$$\begin{aligned}\left\|\widehat{\mathcal{C}}_{YX}-\widetilde{\mathcal{C}}_{YX}\right\|_{\text{op}}\left\|\widehat{\mathcal{C}}_{XX}^{-1}\right\|_{\text{op}} &\leq \gamma^{-1}\left\|\widehat{\mathcal{C}}_{YX}-\widetilde{\mathcal{C}}_{YX}\right\|_{\text{HS}} \\ &\leq \frac{1}{\gamma n}\left(\boldsymbol{\Upsilon}+\boldsymbol{\Upsilon}'\right),\end{aligned}$$

which proves the theorem. □

We now derive approximate kernel versions of two basic probability rules:

$$\text{Sum Rule: } Q(X)=\int_{\Omega} p(X|Y)d\pi(Y) \tag{4.4.17}$$

$$\text{Chain Rule: } Q(X,Y)=p(X|Y)\pi(Y), \tag{4.4.18}$$

where $\pi(Y)$ represents the prior distribution over the random variable Y. We provide minimal details in these derivations in the interest of brevit; please see [98] for a more complete picture.

To get the kernel sum rule, we have that $\mu_X^{\pi}=\mathbb{E}_X\left[\psi(X)\right]=\mathbb{E}_Y\mathbb{E}_{X|Y}\left[\psi(X)\right]$. Therefore,

$$\begin{aligned}\mu_X^{\pi} &= \mathbb{E}_Y\left[\mathcal{C}_{X|Y}\psi(Y)\right] \\ &= \mathcal{C}_{X|Y}\mathbb{E}_Y\left[\psi(Y)\right] \\ &= \mathcal{C}_{X|Y}\mu_X^{\pi}.\end{aligned} \tag{4.4.19}$$

The input μ_Y^{π}, the embedding for $\pi(Y)$ is mapped to μ_X^{π}, the embedding for $Q(X)$, by an application of the conditional embedding operator $\mathcal{C}_{X|Y}$ [98]. The tensor product feature $\psi(x)\otimes\psi(x)$ can also be used to embed the distribution $Q(X)$, resulting in

$$\begin{aligned}\mathcal{C}_{XX}^{\pi} &:= \mathbb{E}_X\left[\psi(x)\otimes\psi(x)\right] \\ &= \mathcal{C}_{XX|Y}^{\pi}\mu_Y^{\pi},\end{aligned} \tag{4.4.20}$$

where the conditional embedding operator $\mathbb{E}_{X|y}\left[\psi(x)\otimes\psi(x)\right]=\mathcal{C}_{XX|Y}^{\pi}\psi(y)$ for tensor product features is used. The reduced-set approximation to the kernel sum rule

Algorithm 8 Reduced-Set Kernel Sum Rule (Abstract)

Input: Initial data $\mathcal{D}_Y = \{\acute{y}_1, \ldots, \acute{y}_n\} \sim \pi(Y)$, $\mathcal{D}_{XY} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim p(X, Y)$, and noise parameter ω^2.

Procedure:

1) Use RSDE on $\mathcal{D}_{XY}$ to get $W = \{w_1, \ldots, w_m\}$, $\mathrm{C} = \{c_1, \ldots, c_m\}$, and $\mathrm{D} = \{d_1, \ldots, d_m\}$.

2) Use RSDE on $\mathcal{D}_Y$ to get $\acute{\mathcal{W}} = \{\acute{w}_1, \ldots, \acute{w}_m\}$ and $\acute{\mathrm{D}} = \{\acute{d}_1, \ldots, \acute{d}_m\}$.

3) Compute $\left(\widetilde{G} + \omega^2 I\right)^{-1}$, where $G_{ij} := k(y_i, y_j)$.

3) Compute feature-mapped data $\Phi := (\psi(c_1), \ldots, \psi(c_m))^T$, $\Psi := (\psi(d_1), \ldots, \psi(d_m))^T$ and $\acute{\Psi} := (\psi(\acute{d}_1), \ldots, \psi(\acute{d}_m))^T$.

4) Compute mean map as

$$\widetilde{\mu}_Y^{\pi} = \acute{\Psi}\alpha, \tag{4.4.21}$$

where $\alpha = \acute{\mathcal{W}}/n$.

4) Compute conditional covariance operator as

$$\widetilde{\mathcal{C}}_{X|Y} = \Phi \left(\widetilde{G} + \omega^2 I\right)^{-1} \Psi. \tag{4.4.22}$$

Output:
Compute conditional mean as

$$\widetilde{\mu}_Y^{\pi} = \widetilde{\mathcal{C}}_{X|Y} \widetilde{\mu}_X^{\pi} = \Phi \left(\widetilde{G} + \omega^2 I\right)^{-1} \acute{G}\alpha, \tag{4.4.23}$$

where $\acute{G}_{ij} := k(d_i, \acute{d}_j)$, and compute the covariance operator as

$$\widetilde{\mathcal{C}}_{XX}^{\pi} = \Phi \,\mathrm{diag}\left(\left(\widetilde{G} + \omega^2 I\right)^{-1} \acute{G}\alpha\right) \Phi^T. \tag{4.4.24}$$

in the RKHS can be computed as in Algorithm 8. Projections onto the reduced-set approximations $\widetilde{\mu}_X^{\pi}$ and $\widetilde{\mathcal{C}}_{XX}^{\pi}$ can be computed using the ideas presented in this section.

In the chain rule, the joint distribution is factorized into the product of a conditional and marginal distribution, as $Q(X, Y) = p(X|Y)\pi(Y)$. The embedding of

Algorithm 9 Reduced-Set Kernel Chain Rule (Abstract)

Input: Initial data $\mathcal{D}_Y = \{\acute{y}_1, \ldots, \acute{y}_n\} \sim \pi(Y)$, $\mathcal{D}_{XY} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim p(X, Y)$, and noise parameter ω^2.

Procedure:

1) Use RSDE on $\mathcal{D}_{XY}$ to get $W = \{w_1, \ldots, w_m\}$, C $= \{c_1, \ldots, c_m\}$, and D $= \{d_1, \ldots, d_m\}$.

2) Use RSDE on $\mathcal{D}_Y$ to get $\acute{\mathcal{W}} = \{\acute{w}_1, \ldots, \acute{w}_m\}$ and $\acute{\text{D}} = \{\acute{d}_1, \ldots, \acute{d}_m\}$.

3) Compute $\left(\widetilde{G} + \omega^2 I\right)^{-1}$, where $G_{ij} := k(y_i, y_j)$.

3) Compute feature-mapped data $\Phi := (\psi(c_1), \ldots, \psi(c_m))^T$, $\Psi := (\psi(d_1), \ldots, \psi(d_m))^T$ and $\acute{\Psi} := (\psi(\acute{d}_1), \ldots, \psi(\acute{d}_m))^T$.

4) Compute Gram matrix $\acute{G}$ as $\acute{G}_{ij} := k(d_i, \acute{d}_j)$.

Output:
Compute cross-covariance operator as

$$\widetilde{\mathcal{C}}^{\pi}_{XY} = \widetilde{\mathcal{C}}_{X|Y}\widetilde{\mathcal{C}}^{\pi}_{YY} = \Phi\left(\widetilde{G} + \omega^2 I\right)^{-1} \acute{G}\,\mathrm{diag}(\alpha)\acute{\Psi}. \tag{4.4.26}$$

$Q(X, Y)$ can be factorized as

$$\begin{aligned}\mathcal{C}^{\pi}_{XY} &= \mathbb{E}_Y\left[\mathbb{E}_{X|Y}\left[\psi(X)\right] \otimes \psi(Y)\right] \\ &= \mathcal{C}_{X|Y}\mathbb{E}_Y\left[\psi(X) \otimes \psi(Y)\right] \\ &= \mathcal{C}_{X|Y}\mathcal{C}^{\pi}_{YY},\end{aligned} \tag{4.4.25}$$

where $\mathcal{C}^{\pi}_{YY}$ is the covariance operator associated to $\pi(Y)$. The kernel chain rule can then be computed as in Algorithm 9, with the projections onto the operators being computed as before.

The kernel sum and chain rules can be used to derive a kernel version of Bayes' rule. Normally, Bayes' rule can be written as $Q(Y|x) = \frac{p(x|Y)\pi(Y)}{Q(x)}$, where $Q(x) = \int_\Omega p(x|Y)d\pi(Y)$, where $p(x|Y)$ is the likelihood function. To derive kernel Bayes' rule, the embedding of $Q(Y|x)$ is constructed via a conditional embedding operator as $\mu^{\pi}_{Y|x} = \mathcal{C}^{\pi}_{Y|x}\psi(x)$. This conditional embedding operator is modified by the prior distribution as

$$\mu^{\pi}_{Y|x} = \mathcal{C}^{\pi}_{Y|x}\psi(x) = \mathcal{C}^{\pi}_{YX}\left(\mathcal{C}^{\pi}_{XX}\right)^{-1}\psi(x), \tag{4.4.27}$$

Algorithm 10 Reduced-Set Kernel Bayes' Rule (Abstract)

Input: Initial data $\mathcal{D}_Y = \{\acute{y}_1, \ldots, \acute{y}_n\} \sim \pi(Y)$, $\mathcal{D}_{XY} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim p(X, Y)$, and noise parameter ω^2.

Procedure:

1) Use RSDE on $\mathcal{D}_{XY}$ to get $W = \{w_1, \ldots, w_m\}$, C $= \{c_1, \ldots, c_m\}$, and D $= \{d_1, \ldots, d_m\}$.

2) Use RSDE on $\mathcal{D}_Y$ to get $\acute{\mathcal{W}} = \{\acute{w}_1, \ldots, \acute{w}_m\}$ and $\acute{\text{D}} = \{\acute{d}_1, \ldots, \acute{d}_m\}$.

3) Compute $\left(\widetilde{G} + \omega^2 I\right)^{-1}$, where $G_{ij} := k(y_i, y_j)$.

3) Compute feature-mapped data $\Phi := (\psi(c_1), \ldots, \psi(c_m))^T$, $\Psi := (\psi(d_1), \ldots, \psi(d_m))^T$ and $\acute{\Psi} := (\psi(\acute{d}_1), \ldots, \psi(\acute{d}_m))^T$.

4) Compute Gram matrix $\acute{G}$ as $\acute{G}_{ij} := k(d_i, \acute{d}_j)$.

5) Compute

$$\Lambda := \left(\widetilde{G} + \omega^2 I\right)^{-1} \acute{G}\,\text{diag}(\alpha) \tag{4.4.30}$$

$$D := \text{diag}\left(\left(\widetilde{G} + \omega^2 I\right)^{-1} \acute{G}\alpha\right), \tag{4.4.31}$$

where $\alpha = \acute{\mathcal{W}}/n$.

Output:
Compute kernel Bayes' rule as

$$\begin{aligned}\widetilde{\mu}^{\pi}_{Y|x} &= \widetilde{\mathcal{C}}^{\pi}_{YX}\left(\left(\widetilde{\mathcal{C}}^{\pi}_{XX}\right)^2 + \tilde{\omega}^2 I\right)^{-1} \widetilde{\mathcal{C}}^{\pi}_{XX}\psi(x) \\ &= \acute{\Psi}\Lambda^T\left((DK)^2 + \tilde{\omega}^2 I\right)^{-1} KDk_x. \end{aligned} \tag{4.4.32}$$

where

$$\mathcal{C}^{\pi}_{XX} := \mathcal{C}^{\pi}_{XX|Y}\mu^{\pi}_Y, \text{ from the sum rule,} \tag{4.4.28}$$

$$\mathcal{C}^{\pi}_{Y|x} := \mathcal{C}_{X|Y}\mathcal{C}^{\pi}_{YY}, \text{ from the chain rule,} \tag{4.4.29}$$

given the embeddings μ_Y and $\mathcal{C}_{YY}$ using the features $\psi(y)$ and $\psi(y)\otimes\psi(y)$ respectively. Putting it all together, kernel Bayes' rule can be computed as in Algorithm 10.

4.5 Experiments

This section reports experimental results for some of the algorithms discussed in this chapter.

4.5.1 Gaussian Process Regression

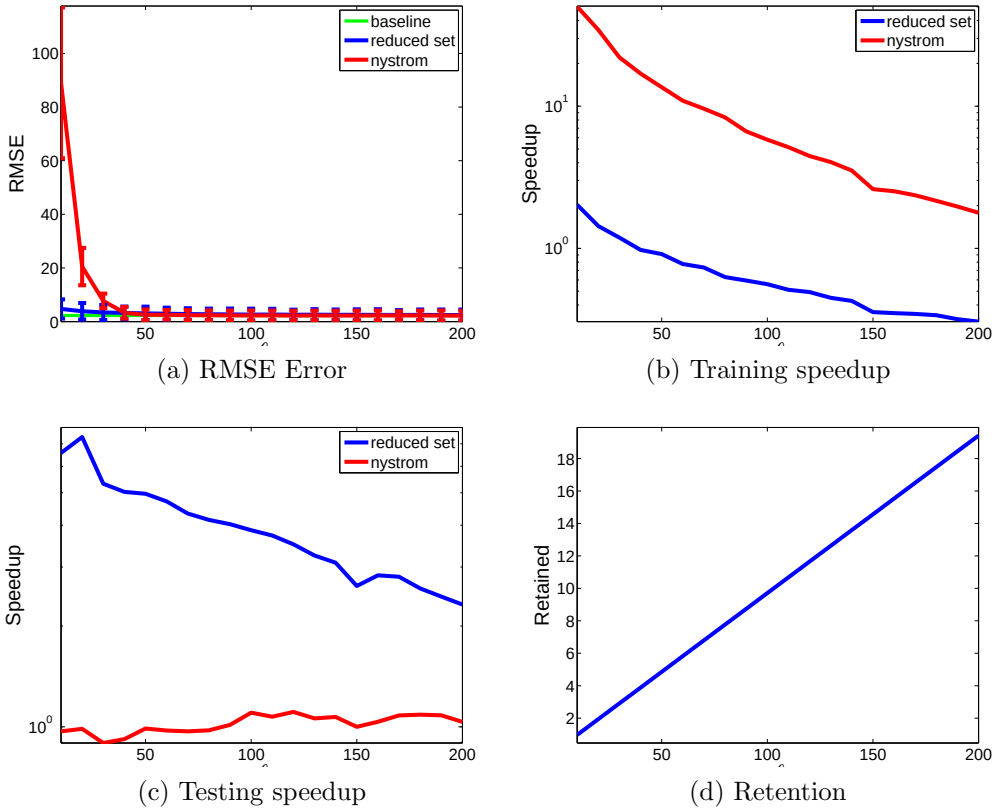
We present results on Gaussian process regression on the datasets given in Table 4. We compare reduced-set approximations to GPR using the k-means algorithm to the Nyström method. The results can be seen in Figures 16 and 17. In the **concrete** dataset, the Nyström method is extremely unstable in terms of RMSE for low values of m, whereas the reduced-set approximation is relatively stable. In the **abalone** case, the Nyström method performs well throughout, although the reduced-set approximation converges to its performance as the number of centers grows. In either case, since the reduced-set method uses k-means, it has a longer training time than Nyström, although it gains a significant testing speedup over Nyström. Therefore the reduced-set approximation should mainly be used for applications that benefit from both a training and testing speedup.

Table 4: Datasets used for Gaussian process regression.

Dataset	n	dim	σ	λ
concrete	1,030	8	3	0.1
abalone	4,177	8	5	0.1

4.5.2 Diffusion Maps

In this section, we give experimental results for the diffusion map applied to a classical toy example, the swiss roll, shown in Figure 18. This dataset has $1,600$ datapoints, and the diffusion embedding is computed using a bandwidth of 10. The reduced-set mapping, with k-means and the number of centers set to 100, is compared to the original mapping. As can be seen in Figure 19, there is little difference in between the two mappings, even at large time scales, which is an indication of high approximation quality in both the eigenfunctions and the eigenvalues. The combined training and



(a) RMSE Error (b) Training speedup

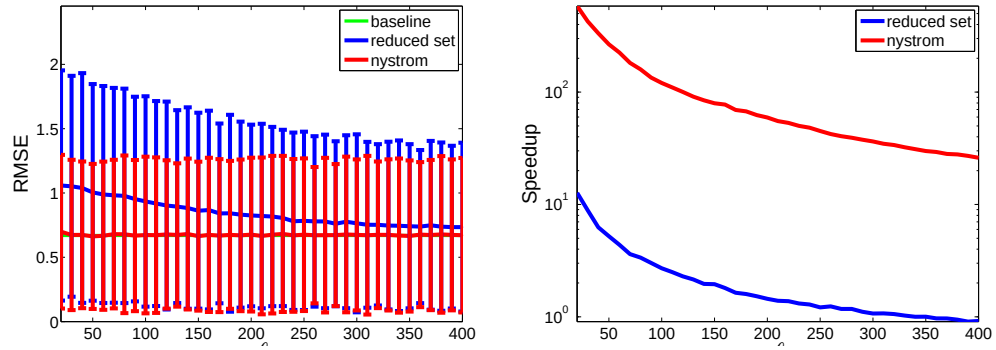
(c) Testing speedup (d) Retention

Figure 16: Gaussian process regression results on the **concrete** dataset.

embedding speedup in this case is about by a factor of 5.

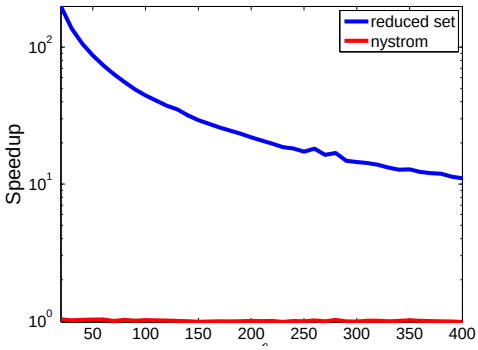
4.5.3 Kernel Embeddings

In this section, we present experimental results for a particular application of kernel embeddings. We apply reduced-set approximations to the kernel Hidden Markov Model, which can be formulated using kernel Bayes' rule, as presented in §4.4. We apply the reduced-set approximation to robot vision data, as outlined in the paper [97]. Here, a video of $2,000$ frames was collected at 6 Hz from a stereo camera mounted on a mobile robot platform circling a stationary obstacle. As in the original experiment, $1,500$ frames were used as training data for each model, and each frame from the training data was reduced to 100 dimensions via SVD on single observations. The goal is to learn a model of the noisy video, and to predict future image observations.

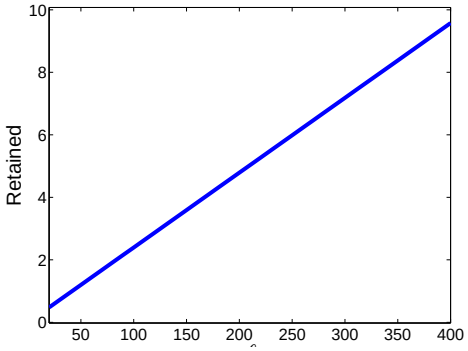


(a) RMSE Error

(b) Training speedup



(c) Testing speedup



(d) Retention

Figure 17: Gaussian process regression results on the **abalone** dataset.

We train a 50-dimensional embedded HMM using Algorithm 1 in [97] with sequences of 20 consecutive observations, its reduced-set approximation, and compare it to a discrete HMM trained using the spectral algorithm presented in [36]. Given a fixed position p in the video, each model is updated online using observations from times $t_1 = 100, 101, \ldots, 250$ from the current position, and then is used to predict an image at times $t_2 = 1, 2, \ldots, 100$ in the future. The results are presented at 100 consecutive postions p near the end of the video. Given the original image I and the predicted image I', the χ^2 distance between their respective histograms is used as a measure of error. The results are presented in Figure 20. Here, we also included results of the unweighted reduced-set approximation, to demonstrate how the weighted version tends to perform slightly better. Overall, while there is a gap in performance between the reduced-set approximation and the full embedding, the approximation performs

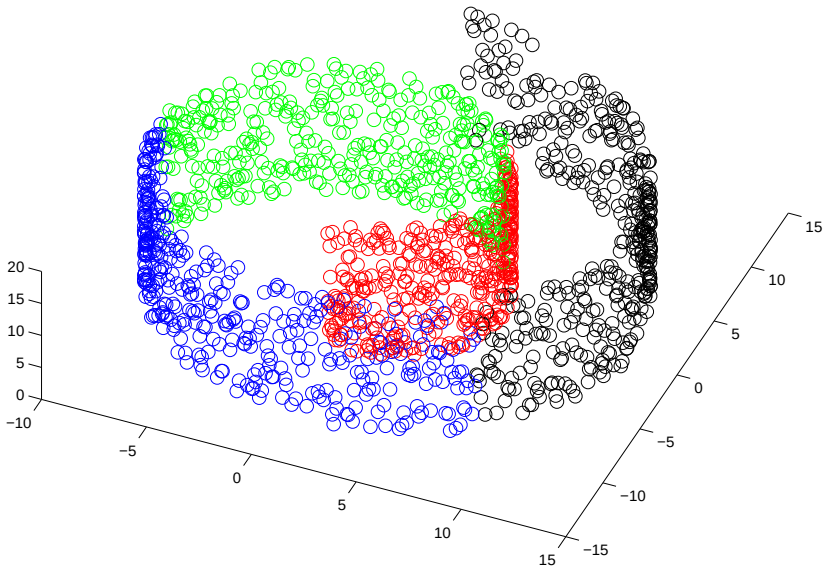
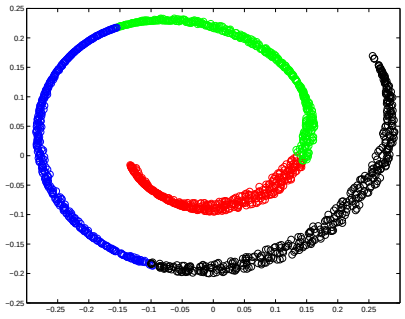


Figure 18: The swiss roll dataset.

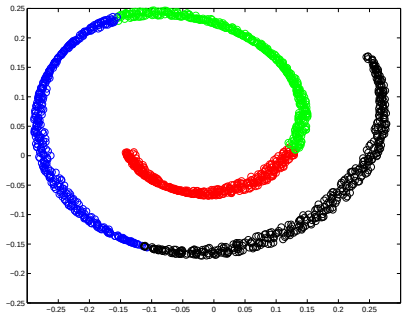
much better than spectral algorithm, and approaches the full set performance as the number of centers increase. Table 5 shows the testing speedup achieved by the approximations, which can be extremely high, even for this relatively small dataset.

Table 5: Testing speedups for RS Kernel HMM over full Kernel HMM for robot vision data.

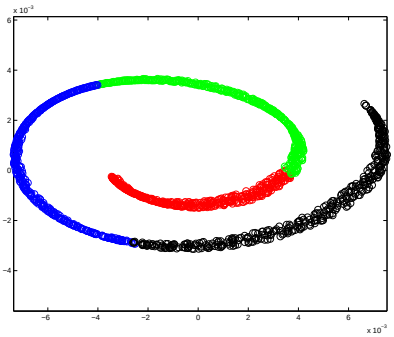
Centers (%age)	Speedup (Mean)	Speedup (Min)	Speedup (Max)
3.13	70	20	180
6.25	45	20	80
15.63	22	13	35
37.50	6.2	4	8.3
56.25	2.5	2.2	2.9



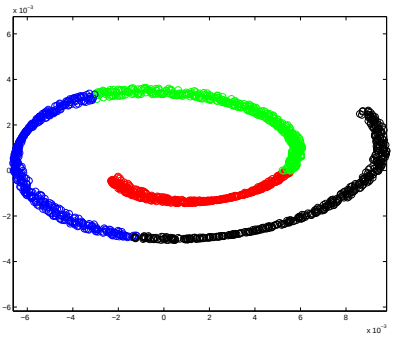
(a) Full set embedding at $t = 1$



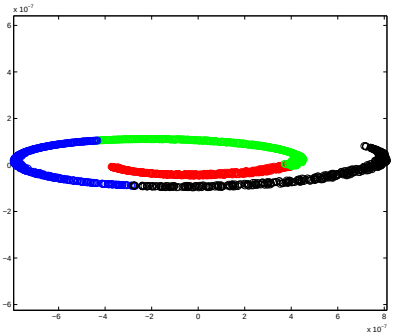
(b) Reduced set embedding at $t = 1$



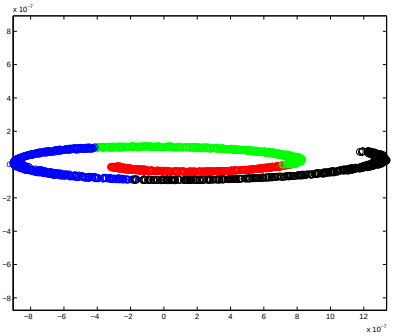
(c) Full set embedding at $t = 5$



(d) Reduced set embedding at $t = 5$



(e) Full set embedding at $t = 15$

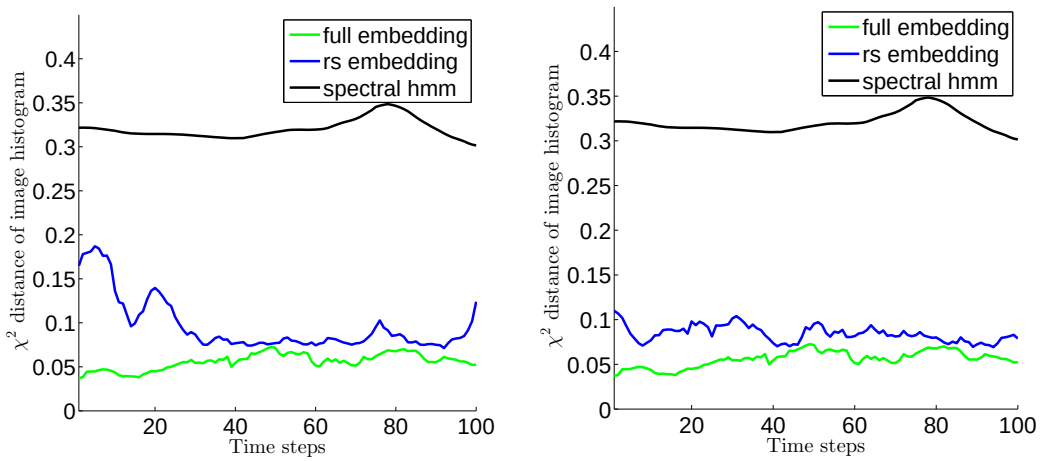


(f) Reduced set embedding at $t = 15$

Figure 19: Diffusion map embeddings at different time scales.

4.6 Conclusion

In this chapter, we showed how to extend the reduced-set kernel PCA algorithm for applications to regression (Gaussian process regression), clustering (diffusion maps),



(a) $m = 3.13\%$ (b) $m = 15.63\%$

Figure 20: Kernel HMM results on different center sets.

and kernel embeddings (kernel Bayes' rule). We proved various theorems bounding the deviation of the reduced-set algorithms from their full-set counterparts, and demonstrated the efficacy of the reduced-set models on various toy and real-world datasets. In summary, it was shown that our presented strategy for reduced-set models for kernel machines in the batch-data case is general enough to be applied to various methods, and effective in practice for speeding up the training and execution speed of these methods at a minimal loss in accuracy.

PART 2: ONLINE DATA

CHAPTER V

REDUCED-SET KERNEL METHODS FOR ADAPTIVE CONTROL

A method to use reduced-set kernel methods for adaptive control is outlined in this chapter. The method, known as Budgeted Kernel Restructuring, is shown to be effective for minimizing tracking error and learning the underlying model uncertainty.

5.1 Introduction

In practice, it is impractical to assume that a perfect model of a system is available for controller synthesis. Assumptions introduced in modeling and changes to system dynamics during operation often result in modeling uncertainties that must be mitigated online. Model-reference adaptive control (MRAC) has been widely studied for broad classes of nonlinear dynamical systems with significant modeling uncertainties [3, 37, 69, 104]. In MRAC, the system uncertainty is approximated using a weighted combination of basis functions, with the numerical values of the weights adapted online to minimize the tracking error. When the structure of the uncertainty is unknown, a neuro-adaptive approach is often employed, in which a neural network (NN) with its weights adapted online is used to capture the uncertainty. A popular example of such a NN is the Gaussian radial-basis-function (RBF) NN, for which the universal-approximation property is known to hold [73]. Examples of MRAC systems employing Gaussian RBFs can be found in [117]. In practice, then, nonlinear MRAC systems typically consist of an additional online-learning component. What differentiates MRAC from traditional online-learning strategies is the existence of the dynamical system and its connection to a control task.

The strong connection between the learning and the evolving dynamical system imposes constraints on the online-learning strategy employed. These constraints ensure the viability of the system's closed-loop stability and performance. For example, classical gradient-based MRAC methods require a condition on persistency of excitation (PE) in the system states [8]. This condition is often impractical to implement and/or infeasible to monitor online. Hence, authors have introduced various modifications to the adaptive law to ensure that the weights stay bounded around an *a-priori* determined value (usually set to 0). Examples of these modifications include Ioannou's σ-mod, Narendra's e-mod, and the use of a projection operator to bound the weights [37, 104]. Recent work in concurrent learning (CL) has shown that if carefully selected online data is used concurrently with current data for adaptation, then the weights remain bounded within a compact neighborhood of the ideal weights [17, 18]. Neither CL nor the aforementioned modifications require PE.

When CL is used, in order to successfully learn the uncertainty, a set of RBF centers must be chosen over the domain of the uncertainty. Prior neuro-adaptive control research assumes that either the centers for the RBF network are fixed [17, 18, 46, 74, 108], or that the centers are moved to minimize the tracking error e [66, 101]. When the centers are fixed, the system designer is assumed to have some domain knowledge about the uncertainty to determine how the centers should be selected. When the centers are moved, the gradient updates favor instantaneous tracking error reduction, and thus the resulting center movement does not guarantee optimal uncertainty capture across the entire operating domain. The work in this chapter creates reduced-set models by utilizing the fact that the Gaussian RBF satisfies the properties of a Mercer kernel. The stucture induced upon RBFs by the Mercer kernel can be used to remove the need for domain knowledge and to propose a novel kernel adaptive control algorithm. RKHS theory is first used to make a connection between PE signals in the state space (e.g., the input space) and PE signals

in the higher-dimensional feature space (with respect to the RBF bases). It is shown that the amount of persistent excitation inserted in the states of the system may vanish or greatly diminish if the RBF centers are improperly chosen. This insight is then utilized along with the kernel-linear-independence test [71] to propose an online algorithm called budgeted kernel restructuring (BKR). BKR picks centers so as to ensure that any excitation in the system does not vanish or diminish. The algorithm is then augmented with CL. The resulting kernel-adaptive algorithm (BKR-CL) uses instantaneous as well as specifically selected data points concurrently for adaptation. Lyapunov-like analysis proves the convergence of the RBF weights to a compact neighborhood of their ideal values if the system states are exciting over a finite interval. BKR-CL does not require PE. Further, BKR-CL can control an uncertain, multivariable dynamical system effectively even if all the RBF centers are initialized to the same value (for example to 0). In addition to removing the assumption of fixed RBF centers, it is shown that, given a fixed budget (maximum number of allowable centers), the algorithm outperforms existing methods that uniformly distribute the centers over an expected domain. The effectiveness of the algorithm is shown using simulations and through comparisons with other neuro-adaptive controllers.

5.2 Preliminaries

The Gaussian function used in RBF networks, expressed as

$$k(x, y) = e^{-\|x-y\|^2/2\mu^2},$$

with bandwidth μ^2, is an example of a bounded kernel function generating an infinite dimensional RKHS $\mathcal{H}$. Fixing a dataset $\mathrm{C} = \{c_1, \ldots, c_m\}$, let

$$\mathcal{F}_{\mathrm{C}} := \left\{ \sum_{i=1}^{m} \alpha_i k(c_i, \cdot) : \alpha_i \in \mathbb{R} \right\}, \tag{5.2.1}$$

where $c_i \in \mathbb{R}^m$, be the *linear subspace generated by* C *in* $\mathcal{H}$. Note that $\mathcal{F}_{\mathrm{C}}$ is a class of functions; any RBF network with the same centers and the same fixed bandwidth

(but without bias) is an element of this subspace. Let $\varphi(x) = [k(x, c_1) \ldots, k(x, c_m)]^T$ and let $W = [w_1, \ldots, w_m]^T$, where $w_i \in \mathbb{R}$. Then $W^T\varphi(x)$ is the output of a standard RBF network. In the machine learning literature, $\sigma(x)$ is sometimes known as the *empirical kernel map*. Two different datasets C_1 and C_2 generate two different families of functions and two different empirical kernel maps $\sigma_{\mathrm{C}_1}(x)$ and $\sigma_{\mathrm{C}_2}(x)$.

Finally, given the above dataset C, one can form an $m \times m$ *kernel matrix* given by $K_{ij} := k(c_i, c_j)$. The following theorem is one of the cornerstones of RBF network theory, and will be used in section IV:

Theorem 5.2.1. ***(Micchelli)****[59] If the function $k(x, y)$ is a positive-definite Mercer kernel, and if all the points in* C *are distinct, the kernel matrix K is nonsingular.*

5.2.1 Persistently Exciting Signals

The work in this thesis also makes use of Tao's definition of a PE signal [104].

Definition 5.2.1. *Let $t \geq 0$ and $T > 0$. Then a bounded vector signal $\Phi(t)$ is* exciting *over an interval $[t, t + T]$ if there exists $\gamma > 0$ such that*

$$\int_t^{t+T} \Phi(\tau)\Phi^T(\tau)d\tau \geq \gamma I. \tag{5.2.2}$$

Definition 5.2.2. *A bounded vector signal $\Phi(t)$ is* persistently exciting *if, for all $t > t_0$, there exist $\gamma > 0$ and $T > 0$ such that*

$$\int_t^{t+T} \Phi(\tau)\Phi^T(\tau)d\tau \geq \gamma I. \tag{5.2.3}$$

The strength of the signal depends on the value of γ.

5.3 Model-Reference Adaptive Control and Concurrent Learning

The formulation of model-reference adaptive control using approximate-model inversion is outlined in this section [17, 40, 41, 45, 66]. Let $\Omega_x \in \mathbb{R}^n$ be compact, $x(t) \in \Omega_x$

be the known state vector, and $u \in \mathbb{R}^k$ denote the control input. Consider the uncertain, multivariable nonlinear dynamical system

$$\dot{x} = f(x(t), u(t)), \tag{5.3.1}$$

where the function f is assumed to be Lipschitz continuous in $x \in \Omega_x$ and the control input u is assumed to be bounded and piecewise continuous. These conditions ensure the existence and uniqueness of the solution to (5.3.1) over a sufficiently large domain Ω_x.

Since the exact model (5.3.1) is usually not available or invertible, an approximate-inversion model $\hat{f}(x, u)$ is used to determine the control input $u = \hat{f}^{-1}(x, \nu)$. Here, ν is the pseudo-control input that represents the desired model output $\dot{x}$ and is expected to be approximately achieved by u. Hence, the pseudo-control input is the output of the approximate-inversion model $\nu = \hat{f}(x, u)$. This approximation results in a model error of the form $\dot{x} = \hat{f}(x, u) + \Delta(x, u)$, where the model error $\Delta : \mathbb{R}^{n+k} \to \mathbb{R}^n$ is given by

$$\Delta(x, u) = f(x, u) - \hat{f}(x, u). \tag{5.3.2}$$

A reference model is chosen to characterize the desired response of the system

$$\dot{x}_{rm}(t) = f_{rm}(x_{rm}(t), r(t)). \tag{5.3.3}$$

The reference-model dynamics are assumed to be continuously differentiable in x for all $x \in \Omega_x \subset \mathbb{R}^n$. It is also assumed that all requirements for guaranteeing the existence of a unique solution to the reference-model equation are satisfied. The command $r(t)$ is assumed to be bounded and piecewise continuous.

Consider a tracking control law ν consisting of a linear feedback part $\nu_{pd} = Kx$, a linear feedforward part $\nu_{rm} = \dot{x}_{rm}$, and an adaptive part $\nu_{ad}(x)$, which can be expressed as

$$\nu = \nu_{rm} + \nu_{pd} - \nu_{ad}. \tag{5.3.4}$$

Define the tracking error e as $e(t) = x_{rm}(t) - x(t)$. Then, letting $A = -K$, the tracking error dynamics are found to be $\dot{e} = Ae + [\nu_{ad}(x,u) - \Delta(x,u)]$. The baseline full-state-feedback controller $\nu_{pd} = Kx$ is designed to ensure that A is a Hurwitz matrix. Hence, for any positive-definite matrix $Q \in \mathbb{R}^{n\times n}$, a positive-definite solution $P \in \mathbb{R}^{n\times n}$ exists to the Lyapunov equation $A^TP+PA+Q = 0$. Let $\bar{x} = [x,u] \in \mathbb{R}^{n+k}$. Assume that it is only known that the uncertainty $\Delta(\bar{x})$ is continuous, and defined over a compact domain $\Omega \subset \mathbb{R}^{n+k}$. Let $\varphi(\bar{x}) = [1, \varphi_2(\bar{x}), \varphi_3(\bar{x}), ..., \varphi_m(\bar{x})]^T$ be a vector of known radial-basis functions. For $i = 2, 3, \ldots, m$, let c_i denote the RBF centroid, and let μ denote the RBF bandwidth. Then, for each i, the radial-basis functions are given as $\varphi_i(x) = e^{-\|\bar{x}-c_i\|^2/2\mu^2}$, which can also be written as $k(c_i, \cdot)$ as per the notation introduced earlier (§5.2). Appealing to the universal-approximation property of RBF-NN [73, 102], given a fixed number of radial-basis functions m, there exist ideal weights $W^* \in \mathbb{R}^{n\times m}$ and a real number $\tilde{\epsilon}(\bar{x})$ such that the following approximation holds for all $x \in \Omega$ (where Ω is compact): $\Delta(x) = W^{*T}\varphi(\bar{x})+\tilde{\epsilon}(\bar{x})$, and $\bar{\epsilon} = \sup_{\bar{x}\in\Omega} \|\tilde{\epsilon}(\bar{x})\|$ can be made arbitrarily small when sufficient number of radial-basis functions are used.

To make use of the universal-approximation property, a RBF-NN can be used to approximate the uncertainty in (5.3.2). In this case, the adaptive element takes the form $\nu_{ad}(\bar{x}) = W^T\varphi(\bar{x})$, where $W \in \mathbb{R}^{n\times m}$. The goal is to design an adaptive law such that $W(t) \to W^*$ as $t \to \infty$, and the tracking error e remains bounded. A commonly used update law, which will be referred to here as the *baseline adaptive law*, is defined as [3, 69, 104]

$$\dot{W} = -\Gamma_W \varphi(\bar{x}) e^T PB. \tag{5.3.5}$$

The adaptive law (5.3.5) guarantees that the weights approach their ideal values (W^*) if and only if the signal $\varphi(\bar{x})$ is PE. In absence of PE, without additional modifications such as σ-mod [37], e-mod [67], or projection-based adaptation [104], the law does not guarantee that the weights W remain bounded. The work in [17], however, shows that

if specifically selected recorded data is used concurrently with instantaneous measurements, then the weights approach and stay bounded in a compact neighborhood of the ideal weights, subject to a sufficient condition on the linear independence of the recorded data; PE is not needed. This idea is formalized in the following theorem:

Theorem 5.3.1. *(**Chowdhary et al** [17]) Consider the system in (5.3.1), the control law of (5.3.4), and $\bar{x}(0) \in \Omega$, where Ω is compact. For the j^{th} recorded data point, let $\epsilon_j(t) = W^T(t)\varphi(\bar{x}_j) - \Delta(\bar{x}_j)$, with $\Delta(\bar{x}_j)$ for a stored data point j calculated using (5.3.2) as $\Delta(\bar{x}_j) = \dot{x}_j - \nu_j$. Also, let p be the number of recorded data points $\varphi(\bar{x}_j)$ in the matrix $Z = [\varphi(\bar{x}_1), ..., \varphi(\bar{x}_p)]$. If $\text{rank}(Z) = m$, then the following weight-update law*

$$\dot{W} = -\Gamma_W \varphi(\bar{x}) e^T P B - \Gamma_W \sum_{j=1}^{p} \varphi(\bar{x}_j)\epsilon_j^T, \tag{5.3.6}$$

renders the tracking error e and the RBF-NN weight errors $\tilde{W}$ uniformly ultimately bounded. Furthermore, the adaptive weights $W(t)$ will approach and remain bounded in a compact neighborhood of the ideal weights W^*.

The matrix Z will be referred to as the *history stack*.

5.4 Kernel Linear Independence and the Budgeted Kernel Restructuring (BKR) Algorithm

PE Signals and the RKHS In this section, a connection between adaptive control and kernel methods is made by leveraging RKHS theory to relate PE of $\bar{x}(t)$ to PE of $\varphi(\bar{x}(t))$.

Theorem 5.4.1. *Suppose $\bar{x}(t)$ evolves in the state space according to (5.3.1). If there exists some time $t_f \in \mathbb{R}_+$ such that the mapping $\psi(\bar{x}(t)) \to \mathcal{H}$ for $t > t_f$ is orthogonal to the linear subspace $\mathcal{F}_\text{C} \subset \mathcal{H}$ for all time, then the signal $\varphi(\bar{x}(t))$ is not persistently exciting.*

Proof. Let $\mathrm{C} = \{c_1, \ldots, c_m\}$, and recall that $\mathcal{F}_{\mathrm{C}}$ represents the linear subspace generated by C in $\mathcal{H}$ (see (5.2.1)). Let $G = \varphi(\bar{x}(t))\varphi(\bar{x}(t))^T$.

$$
\begin{aligned}
G &= \begin{pmatrix} k(\bar{x}, c_1) \\ \vdots \\ k(\bar{x}, c_m) \end{pmatrix} \begin{pmatrix} k(\bar{x}, c_1) & \cdots & k(\bar{x}, c_m) \end{pmatrix} \\
&= \begin{pmatrix} k(\bar{x}, c_1)k(\bar{x}, c_1) & \cdots & k(\bar{x}, c_1)k(\bar{x}, c_m) \\ \vdots & \ddots & \vdots \\ k(\bar{x}, c_m)k(\bar{x}, c_1) & \cdots & k(\bar{x}, c_m)k(\bar{x}, c_m) \end{pmatrix} \\
&= \begin{pmatrix} \langle \psi_{\bar{x},c_1} \rangle \langle \psi_{\bar{x},c_1} \rangle & \cdots & \langle \psi_{\bar{x},c_1} \rangle \langle \psi_{\bar{x},c_m} \rangle \\ \vdots & \ddots & \vdots \\ \langle \psi_{\bar{x},c_m} \rangle \langle \psi_{\bar{x},c_1} \rangle & \cdots & \langle \psi_{\bar{x},c_m} \rangle \langle \psi_{\bar{x},c_m} \rangle \end{pmatrix},
\end{aligned}
$$

where $\langle \psi_{\bar{x},c_i} \rangle$ is shorthand for $\langle \psi(\bar{x}), \psi(c_i) \rangle$. A matrix G is positive definite if and only if $v^T G v > 0, \ \forall v \in \mathbb{R}^m$. In the above, this translates to

$$
\begin{aligned}
v^T G v &= \sum_{i,j=1}^{l} v_i v_j G_{i,j} \\
&= \sum_{i,j=1}^{l} v_i v_j \langle \psi(\bar{x}), \psi(c_i) \rangle \langle \psi(\bar{x}), \psi(c_j) \rangle \\
&= \left\langle \psi(\bar{x}), \sum_{i=1}^{l} v_i \psi(c_i) \right\rangle \left\langle \psi(\bar{x}), \sum_{j=1}^{l} v_j \psi(c_j) \right\rangle \\
&= \left\langle \psi(\bar{x}), \sum_{i=1}^{m} v_i \psi(c_i) \right\rangle^2 .
\end{aligned}
$$

It is known that if $c_i \neq c_j$, then $\psi(c_i)$ and $\psi(c_j)$ are linearly independent in $\mathcal{H}$[59]. Further, if the trajectory $\bar{x}(t) \in \mathbb{R}^m$ is bounded for all time, then $k(\bar{x}(t), c) \neq 0$. From this, it follows that the signal $\int_t^{t+T} G(\tau) d\tau$ is bounded. □

Corollary 5.4.1. *Suppose $\bar{x}(t)$ evolves in the state space according to (5.3.1). If there exists some state $\bar{x}_f \in \mathbb{R}^n$ and some $t_f \in \mathbb{R}_+$ such that $\bar{x}(t) = \bar{x}_f \ \forall t > t_f$, then $\varphi(\bar{x}(t))$ is not persistently exciting.*

Proof. Apply the previous theorem when the state $\bar{x}(t)$ reaches a stationary point. □

PE of $\varphi(\bar{x}(t))$ therefore follows only if neither of the above conditions are met. Figure 21 shows an example of a trajectory $\bar{x}(t)$ mapped to $\mathcal{H}$. The signal $\psi(\bar{x}(t))$

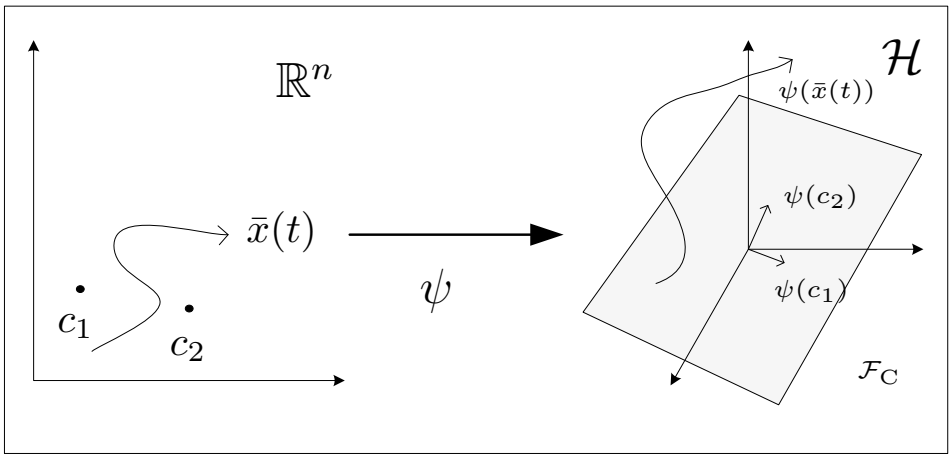


Figure 21: An example of a RKHS space mapping. The trajectory $\bar{x}(t)$ evolves in the state space $\mathbb{R}^n$ via an ODE; the image $\psi(\bar{x}(t))$ is an equivalent ODE in $\mathcal{H}$. If c_1 and c_2 are the centers for the RBFs, then they generate the linear subspace $\mathcal{F}_{\mathrm{C}} \subset \mathcal{H}$, which is a family of linearly parameterized functions.

in $\mathcal{H}$ becomes orthogonal to the centers $C = \{c_1, \ldots, c_m\}$ mapped to $\mathcal{H}$ if $\bar{x}(t)$ moves far away from C in $\mathbb{R}^n$. Although orthogonality is desired in the state space $\mathbb{R}^n$ for guaranteeing PE, orthogonality in $\mathcal{H}$ is detrimental to PE of $\varphi(\bar{x}(t))$. Recall that the baseline adaptive law of (5.3.5) does not guarantee that the weights converge to a neighborhood of the ideal weights if $\varphi(\bar{x}(t))$ is not persistently exciting. PE of $\varphi(\bar{x}(t))$ requires that (1) $\bar{x}(t)$ is persistently exciting and (2) the centers C are chosen such that the image $\psi(\bar{x}(t)) \in \mathcal{H}$ is not orthogonal to the linear subspace $\mathcal{F}_{\mathrm{C}}$ generated by the centers. If the system evolves far away from the original set of centers, then the old centers will give less accurate information than new ones selected to better represent the current operating domain. In other words, keeping the centers close to $\bar{x}(t)$ maintains PE of $\varphi(\bar{x}(t))$ as long as $\bar{x}(t)$ is also persistently exciting. At the same time, all of the centers selected earlier cannot be discarded since

it is desirable to preserve global function approximation over the entire domain. The design of a good algorithm for picking centers therefore needs to (1) satisfy the global approximation property and (2) avoid the geometric condition stated in Theorem 5.4.1 that leads to loss of PE. Figure 22 depicts a signal that loses PE with respect to the centers C in the Hilbert space $\mathcal{H}$. The work in [17] shows that if the history stack

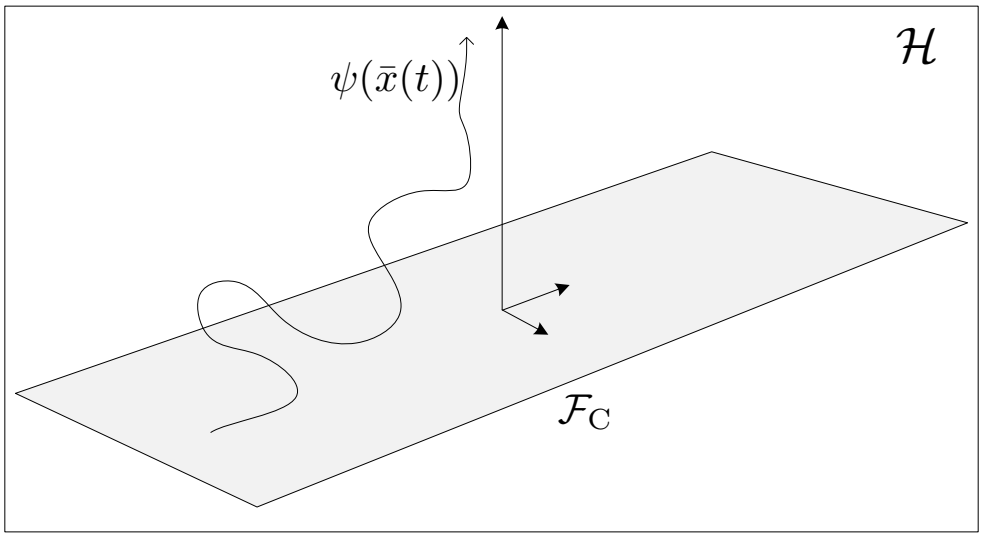


Figure 22: An example of a signal losing PE in ψ. Any trajectory $\bar{x}(t) \in \mathbb{R}^n$ inducing a trajectory $\psi(\bar{x}(t)) \in \mathcal{H}$ that is orthogonal to $\mathcal{F}_{\mathrm{C}}$ after some time t_f causes $\varphi(\bar{x}(t))$ to lose PE.

is linearly independent, the law in (5.3.6) suffices to ensure convergence to a compact neighborhood of the ideal weight vector W^*, without requiring PE of $\varphi(\bar{x}(t))$. The CL gradient descent can be reinterpreted in the RKHS framework. Note that the CL adaptive law in (5.3.6) will ensure $\epsilon_j(t) \to \tilde{\epsilon}$ by driving $W(t) \to W^*$. By the above analysis, $\sigma(\bar{x}_j)$ is the projection of $\psi(\bar{x}_j)$ onto $\mathcal{F}_{\mathrm{C}}$. If $\psi(\bar{x}(t))$ is orthogonal to $\mathcal{F}_{\mathrm{C}}$ in $\mathcal{H}$, the first term in (5.3.6) (which is also the baseline adaptive law of (5.3.5)), vanishes, causing the evolution of weights to stop with respect to the tracking error e. The condition that $\sum_p \varphi(\bar{x}_j)\varphi(\bar{x}_j)^T$ has the same rank as the number of centers in the RBF network is equivalent to the statement that one has a collection of vectors $\{\bar{x}_j\}_{j=1}^p$ where the projections of $\{\psi(\bar{x}_j)\}_{j=1}^p$ onto $\mathcal{F}_{\mathrm{C}}$ allow the weights to continue

evolving to minimize the difference in uncertainty.

Linear Independence A strategy to select the RBF centers online can be chosen using Theorems 5.4.1 and 5.4.1. The algorithm presented here ensures that the centers cover the current domain of operation as well as keeping at least some centers that reflect previous domains of operation. The scheme introduced in [71] can be used to ensure that the RBF centers reflect the current domain of operation. Similar ideas are used in the kernel-adaptive filtering literature to curb filter complexity [57]. The algorithm maintains a 'dictionary' of centers $\mathrm{C}_m = \{c_i\}_{i=1}^m$, where m is the current size of the dictionary, and N_D is the upper limit on the number of points (the budget). A new center $c_{m'}$ is inserted into the dictionary based on whether $c_{m'}$ can be approximated in $\mathcal{H}$ by the current set of centers C_m. The ability of C_m to approximate $c_{m'}$ is checked using

$$\gamma = \min_{a_i} \left\| \sum_{i=1}^{m} a_i \psi(c_i) - \psi(c_{m'}) \right\|_{\mathcal{H}}^2, \tag{5.4.1}$$

where the a_i denote the coefficients of the linear independence test. Unraveling (5.4.1) in terms of (2.2.3) shows that the coefficients a_i can be determined by minimizing γ, which yields the optimal coefficient vector $\hat{a}_m = K_m^{-1}\hat{k}_m$, where $K_m = k(\mathrm{C}_m, \mathrm{C}_m)$ is the kernel matrix for the dictionary dataset C_m, and $\hat{k}_m = k(\mathrm{C}_m, c_{m'})$ is the kernel vector. Substituting the optimal value $\hat{a}_m$ into (5.4.1) yields

$$\gamma = k(c_{m'}, c_{m'}) - \hat{k}_m^T \hat{a}_m. \tag{5.4.2}$$

The algorithm is summarized in Algorithm 11. For more details see [71]; in particular, an efficient way to implement the updates is given there.

Note that due to the nature of the linear independence test, every time a state $\bar{x}(t)$ is encountered that cannot be approximated within tolerance η by the current dictionary C_m, it is added to the dictionary. Further, since the dictionary is designed to keep the most varied basis possible, this is the best one can do on a budget using

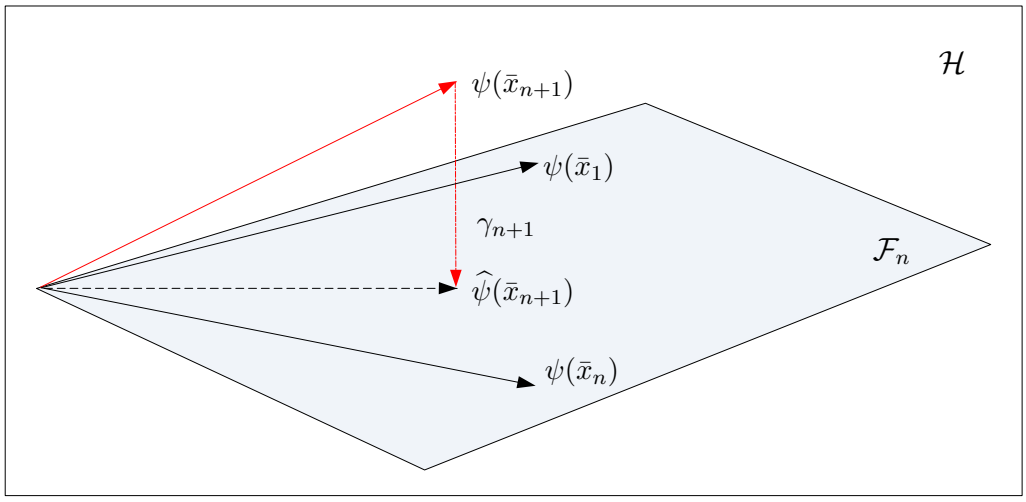


Figure 23: An example of the projection of $\psi(\bar{x}_{n+1})$ onto the subspace $\mathcal{F}_n$ spanned by $\{\psi(\bar{x}_1), \ldots, \psi(\bar{x}_n)\}$. The scalar γ_{n+1} is the length of the residual, and is a measure of the independence of $\psi(\bar{x}_{n+1})$ w.r.t. $\mathcal{F}_n$.

only instantaneous online data. Therefore, in the final algorithm, all the centers can be initialized to 0, and the linear independence test can be periodically run to check whether or not to add a new center to the RBF network. This is fundamentally different from update laws of the kind given in [66], which attempt to move the centers to minimize the tracking error e. Since such update laws are essentially rank-1, if all the centers are initialized to 0, they will all move in the same direction together. Furthermore, from the above analysis, it is clear that the amount of excitation is maximized when the centers are previous states themselves.

Summarizing, the following are the advantages of picking centers online with the linear independence test (5.4.2):

1. In light of Theorem 5.4.1, Algorithm 11 ensures inserted excitation at the current time in the system does not disappear by selecting centers such that $\mathcal{F}_{\mathrm{C}}$ is not orthogonal to current states. In less formal terms, this means that at least some centers are "sufficiently" close to the current state.

2. This also implies that not all of the old centers are discarded, which means that

all of the regions where the system has previously operated are represented to some extent in the dictionary. This implies global function approximation.

3. Algorithm 11 enables the design of adaptive controllers without any prior knowledge of the domain. For example, this method would allow one to initialize all centers to zero, with appropriate centers then selected by Algorithm 11 online.

4. On a budget, selecting centers with Algorithm 11 is better than evenly spacing centers, since the centers are selected along the path of the system in the state space. This results in a more judicious distribution of centers without any prior knowledge of the uncertainty.

If the centers for the system are picked using the kernel-linear-independence test, and the weight update law is given by the standard baseline law (5.3.5), the resulting algorithm is called budgeted kernel restructuring (BKR).

Algorithm 11 Kernel-Linear-Independence Test

Input: New point $c_{m'}, \eta, N_D$.
Compute
$\hat{a}_m = K_l^{-1}\hat{k}_m$
$k_l = k(\mathrm{C}_m, c_{m'})$
Compute γ as in (5.4.2)
if $\gamma > \eta$ **then**
  if $l < N_D$ **then**
    Update dictionary by storing the new point $c_{m'}$, and recalculating γ's for each of the points
  else
    Update dictionary by deleting the point with the minimal γ, and then recalculate γ's for each of the points
  end if
end if

5.5 The BKR-CL Algorithm

Motivation The BKR algorithm selects centers in order to ensure that any inserted excitation does not vanish. However, it must be proven that the weights are driven

towards their ideal values, in order to fully leverage the universal-approximation property. In this section, a CL adaptive law that guarantees that the weights approach and remain bounded in a compact domain around their ideal weights is presented. The law concurrently utilizes recorded data with instantaneous data for adaptation, and is wedded with BKR to create BKR-CL.

First, the algorithm used to select, record, and remove data points in the history stack is described. Note that Algorithm 11 picks the centers discretely; therefore, there always exists an interval $[t_k, t_{k+1}]$ where $k \in \mathbb{N}$, and the centers are fixed. This discrete update of the centers introduces switching in the closed loop system. Let $\varphi^k(\bar{x})$ denote the value of φ given by this particular set of centers, denote by W^{k^*} the ideal set of weights for these centers, and by φ^k the radial basis function associated to these centers. Let $p \in \mathbb{N}$ denote the subscript of the last point stored. For a stored data point $\bar{x}_j$, let $\varphi_j^k \in \mathbb{R}^n$ denote $\varphi^k(\bar{x}_j)$. If $S_t = [\bar{x}_1, \ldots, \bar{x}_p]$ denotes the matrix containing the recorded information in the history stack at time t, then $Z_t = [\varphi_1^k, \ldots, \varphi_p^k]$ is the matrix containing the output of the RBF function for the current set of RBF centers. The p-th column of Z_t will be denoted by $Z_t(:, p)$. It is assumed that the maximum allowable number of recorded data points is limited due to memory or processing power considerations. Therefore, Z_t is constrained to have a maximum of $p \in \mathbb{N}$ columns; clearly, in order to be able to satisfy $\text{rank}(Z_t) = m$, $p \geq m$. For the j-th data point, the associated model error $\Delta(\bar{x}_j)$ is assumed to be stored in the array $\bar{\Delta}(:, j) = \Delta(\bar{x}_j)$. The uncertainty of the system is estimated by estimating $\dot{x}_j$ for the j^{th} recorded data point using optimal fixed point optimal smoothing, and solving equation (5.3.2) to get $\Delta(\bar{x}) = \dot{x}_j - \nu_j$.

The history stack is populated using an algorithm that aims to maximize the minimum singular value of the symmetric matrix $\Omega = \sum_{j=1}^{p} \varphi^k(x_j)\varphi^{k^T}(x_j)$ (see Algorithm 12). Thus any data point linearly independent of the data stored in the history

stack, is included in the history stack. At the initial time $t = 0$ and $k = 0$, the algorithm begins by setting $Z_t(:, 1) = \varphi^0(\bar{x}(t_0))$. The algorithm then selects sufficiently different points for storage; a point is considered sufficiently different if it is linearly independent of the points in the history stack, or if it is sufficiently different in the sense of the Euclidean norm of the last point stored. Furthermore, if Algorithm 11 updates the dictionary by adding a center, then this point is also added to the history stack (if the maximum allowable size of the history stack is not reached), so the rank of Z_t is maintained. If the number of stored points exceeds the maximum allowable number, the algorithm seeks to incorporate new data points in manner that the minimum singular value of Z_t is increased; the current data point is added to the history stack only if swapping it with an old point increases the singular value. One can see that Algorithm 11 and Algorithm 12 work similarly to one another. The structure of Algorithm 12 is a condition that relates to the PE of the signal $\bar{x}(t)$, by ensuring that the history stack is as linearly independent in $\mathbb{R}^n$ as possible, while Algorithm 11 does the same for the PE of the signal $\varphi(\bar{x}(t))$ as well, trying to ensure that the center set is as linearly independent in $\mathcal{H}$ as possible. In order to approximate the uncertainty well, both algorithms are needed.

Lyapunov Analysis In this section, a result proving the boundedness of the closed-loop signals when using BKR-CL is reported. Over each interval, between switches in the NN approximation, the tracking-error dynamics are given by the following differential equation

$$\dot{e} = Ae + [W^T \varphi^k(\bar{x}) - \Delta(\bar{x})]. \tag{5.5.1}$$

The NN-approximation error for the k^{th} system can be rewritten as

$$\Delta(\bar{x}) = W^{k^{*T}} \varphi^k(\bar{x}) + \tilde{\epsilon}^k(\bar{x}), \tag{5.5.2}$$

An equivalent of Theorem 5.3.1 for the following switching weight-update law is proven:

$$\dot{W} = -\Gamma_W \varphi^k(\bar{x}) e^T P B - \Gamma_W \sum_{j=1}^{p} \varphi^k(\bar{x}_j) \epsilon_j^{k^T}. \tag{5.5.3}$$

Theorem 5.5.1. *Consider the system in (5.3.1), the control law of (5.3.4), and* $\bar{x}(0) \in \Omega$ *where* Ω *is compact. For the* j^{th} *recorded data point let* $\epsilon_j^k(t) = W^T(t)\varphi^k(\bar{x}_j) - \Delta(\bar{x}_j)$ *and let* p *be the number of recorded data points* $\varphi_j^k := \varphi^k(\bar{x}_j)$ *in the matrix* $Z = [\varphi_1^k, \ldots, \varphi_p^k]$*, such that* $\text{rank}(Z_0) = m$ *at* $t = 0$*. Assume that the RBF centers are updated using Algorithm 11, and the history stack is updated using Algorithm 12. Then, the weight-update law in (5.5.3) ensures that the tracking error* e *of (5.5.1) and the RBF-NN weight errors* $\tilde{W}^k$ *of (5.5.2) are bounded.*

Proof. Consider the tracking error dynamics given by (5.5.1) and the update law of (5.5.3). Since $\nu_{ad} = W^T(\sigma^k(x))$, we have $\epsilon_j = W^T\sigma^k(\bar{x}) - \Delta(\bar{x})$. Then the NN approximation error is now given by (5.5.2). With $\tilde{W}^k = W - W^{k^*}$

$$\begin{aligned}\epsilon_j^k(\bar{x}) &= W^T\sigma^k(\bar{x}) - W^{k^{*^T}}\sigma^k(\bar{x}) - \tilde{\epsilon}^k(\bar{x}) \\ &= \tilde{W}^k\sigma^k(\bar{x}) - \tilde{\epsilon}^k(\bar{x}).\end{aligned}$$

Therefore over $[t_k, t_{k+1}]$, the weight dynamics are given by the following switching system

$$\dot{\tilde{W}}^k(t) = -\Gamma\left[\left(\sum_{j=1}^{p}\sigma^k(\bar{x}_j)\sigma^{k^T}(\bar{x}_j)\right)\tilde{W}^k(t) + \sum_{j=1}^{p}\sigma^k(\bar{x}_j)\tilde{\epsilon}^{k^T}(z_j) - \sigma^k(\bar{x}(t))e^T(t)PB\right]. \tag{5.5.4}$$

Consider the family of positive definite functions $V^k = \frac{1}{2}e^TPe + \frac{1}{2}\,\text{Tr}\left(\tilde{W}^{k^T}\Gamma_W^{-1}\tilde{W}^k\right)$, where $\text{Tr}(\cdot)$ denotes the trace operator. Note that $V^k \in \mathscr{C}^1$, $V^k(0) = 0$ and

$V^k(e, \tilde{W}^k) > 0 \ \forall e, \tilde{W}^k \neq 0$, and define $\Omega^k := \sum_j \sigma_j^k s_j^{k^T}$. Then

$$\begin{aligned}
\dot{V}^k(e, \tilde{W}^k) &= -\frac{1}{2} e^T Q e + e^T P B \tilde{\epsilon}^k - \text{Tr}\left(\tilde{W}^{k^T} \left[\sum_j \sigma_j^k \sigma_j^{kT} \tilde{W}^k - \sum_j \sigma_j \tilde{\epsilon}_j^k \right] \right) \\
&\leq -\frac{1}{2} \lambda_{\min}(Q) e^T e + \left\| e^T P B \tilde{\epsilon}^k \right\| - \lambda_{\min}(\Omega^k) \tilde{W}^{k^T} \tilde{W}^k - \left\| \tilde{W}^{k^T} \sum_i \sigma_i^k \tilde{\epsilon}^k \right\| \\
&\leq \|e\| \left(C_1^k - \frac{1}{2} \lambda_{\min}(Q) \|e\| \right) + \left\| \tilde{W}^k \right\| \left(C_2^k - \lambda_{\min}(\Omega^k) \left\| \tilde{W}^k \right\| \right),
\end{aligned}$$

where $C_1^k = \|PB\| \tilde{\epsilon}^k$, $C_2 = p \tilde{\epsilon}^k \sqrt{l}$ (l being the number of RBFs in the network), and $\Omega^k = \sum_{j=1}^{p} \sigma^k(\bar{x}_j) \sigma^{k^T}$. Note that since Algorithm 12 guarantees that only sufficiently different points are added to the history stack then due to Micchelli's theorem [59], Ω^k is guaranteed to be positive definite. Hence if $\|e\| > 2C_1/\lambda_{\min}(Q)$ and $\|\tilde{W}^k\| > C_3/\lambda_{\min}(\Omega^k)$, we have $\dot{V}(e, \tilde{W}^k) < 0$. Hence the set $\Pi^k = \{(e, \tilde{W}^k) : \|e\| + \|\tilde{W}^k\| \leq 2C_1/\lambda_{\min}(Q) + C_2/\lambda_{\min}(\Omega^k)\}$ is positively invariant for the k^{th} system.

Let $\mathcal{S} = \{(t_1, 1), (t_2, 2), \dots\}$ be an arbitrary switching sequence with finite switches in finite time (note that this is always guaranteed due to the discrete nature of Algorithm 11). The sequence denotes that a system S_k was active between t_k and t_{k+1}. Suppose at time t_{k+1}, the system switches from S_k to S_{k+1}. Then $e(t_k) = e(t_{k+1})$ and $\tilde{W}^{k+1} = \tilde{W}^k + \Delta W^{k^*}$, where $\Delta W^{k^*} = W^{k^*} - W^{(k+1)^*}$. Since $\dot{V}^k(e, \tilde{W}^k)$ is guaranteed to be negative definite outside of a compact interval, it follows that $e(t_k)$ and $\tilde{W}^k(t_k)$ are guaranteed to be bounded. Therefore, $e(t_{k+1})$ and $\tilde{W}^{k+1}(t_{k+1})$ are also bounded and since $\dot{V}^{k+1}(e, \tilde{W}^{k+1})$ is guaranteed to be negative definite outside of a compact set, $e(t)$ and $\tilde{W}^{k+1}(t)$ are also bounded. Furthermore, over every interval $[t_k, t_{k+1}]$, they will approach the positively invariant set Π^{k+1} or stay bounded within Π^{k+1} if inside. □

Remark 1. *Note that BKR-CL along with the assumption that* $\text{rank}(Z) = l$ *guarantees that the weights stay bounded in a compact neighborhood of their ideal values without requiring any additional damping terms such as* σ *modification [37] or e modification [69] (even in the presence of noise). Due to Micchelli's theorem [59], as*

long as the history stack matrix Z_k contains l different points, $\text{rank}(Z) = l$*. Therefore, it is expected that this rank condition will be met within the first few time steps even when one begins with no* a-priori *recorded data points in the history stack. In addition, a projection operator (see for example [104]) can be used to bound the weights until* $\text{rank}(Z_t) = l$*. This is a straightforward extension of the presented approach.*

Algorithm 12 Singular-Value Maximizing Algorithm for Recording Data Points

Require: $p \geq 1$
if $\frac{\|\varphi^k(x(t)) - \varphi_p^k\|^2}{\|\varphi^k(x(t))\|} \geq \epsilon$ **or** a new center added by Algorithm 11 without replacing an old center **then**
  $p = p + 1$
  $S_t(:, p) = x(t)$; {store $\bar{\Delta}(:, p) = \dot{x} - \nu(t)$}
else if new center added by Algorithm 11 by replacing an old center **then**
  if old center was found in S_t **then**
    overwrite old center in the history stack with new center
    set p equal to the location of the data point replaced in the history stack
  end if
end if
Recalculate $Z_t = [\varphi_1^k, \ldots, \varphi_p^k]$
if $p \geq \bar{p}$ **then**
  $T = Z_t$
  $S_{old} = \min \text{SVD}(Z_t^T)$
  for $j = 1$ to p **do**
    $Z_t(:, j) = \varphi^k(x(t))$
    $S(j) = \min \text{SVD}(Z_t^T)$
    $Z_t = T$
  end for
  find $\max S$ and let k denote the corresponding column index
  if $\max S > S_{old}$ **then**
    $S_t(:, k) = x(t)$; {store $\bar{\Delta}(:, k) = \dot{x} - \nu(t)$}
    $Z_t(:, k) = \varphi^k(x(t))$
    $p = p - 1$
  else
    $p = p - 1$
    $Z_t = T$
  end if
end if

5.6 Application to Control of Wing-Rock Dynamics

Modern highly swept-back or delta-wing fighter aircraft are susceptible to lightly-damped oscillations in roll known as 'wing-rock'. Wing-rock often occurs at conditions commonly encountered at landing ([84]). Precision control in presence of wing-rock is critical for safe landing. In this section, BKR-CL is used to to track a sequence of roll commands in the presence of simulated wing-rock dynamics. Let θ denote the roll attitude of an aircraft, p denote the roll rate, and δ_a denote the aileron-control input. Then, a model for wing-rock dynamics is [61]

$$\dot{\theta} = p, \dot{p} = L_{\delta_a}\delta_a + \Delta(x), \tag{5.6.1}$$

where

$$\Delta(x) = W_0^* + W_1^*\theta + W_2^* p + W_3^*|\theta|p + W_4^*|p|p + W_5^*\theta^3 \tag{5.6.2}$$

and $L_{\delta_a} = 3$. The parameters for wing-rock motion are adapted from [93]; they are $W_1^* = 0.2314, W_2^* = 0.6918, W_3^* = -0.6245, W_4^* = 0.0095, W_5^* = 0.0214$. In addition to these parameters, a trim error is introduced by setting $W_0^* = 0.8$. The ideal parameter vector W^* is assumed to be unknown. The chosen inversion model has the form $\nu = \frac{1}{L_{\delta_a}}\delta_a$. This choice results in the modeling uncertainty of (5.3.2) being given by $\Delta(x)$. The adaptive controller uses the control law of (5.3.4). The linear gain K of the control law is given by $[0.5, 0.4]^T$. A second-order reference model with a natural frequency of 1 rad/sec, and damping ratio of 0.5 is chosen, and the learning rate is set to $\Gamma_W = 1$. The simulation uses a time-step of 0.01 sec.

The simulation compares BKR-CL with e-mod, σ-mod and projection-operator adaptive laws (henceforth collectively denoted as 'classical adaptive laws') for neuro-adaptive control. The number of centers was chosen to be 12. In BKR-CL, the centers were all initialized to $0 \in \mathbb{R}^2$ at $t = 0$, while in the classical adaptive laws, they were placed across the domain of uncertainty evenly (where the prior knowledge

of the domain was determined through experiments conducted beforehand). The BKR-CL algorithm uses the switching weight-update law (5.5.3). Figure 24a shows the tracking of the reference model by the classical laws and BKR-CL. Figure 24b shows the tracking error for the same. As can be seen, the overall error, and especially the transient error for BKR-CL, is lower than that for the classical laws. Figure 25 shows an example of how concurrent learning affects the evolution of weights; it has a tendency to spread the weights across a broader spectrum of values than the classical laws, leading the algorithm to choose from a richer class of functions in $\mathcal{F}_C$. Figure 26 shows the final set of centers that are used; as can be seen, the centers follow the path of the system in the state space. This is another reason why the uncertainty is approximated better by BKR-CL than the classical algorithms.

5.6.1 Illustration of long-term learning effect introduced by BKR-CL

One of the main contributions of the presented BKR-CL scheme is the introduction of long-term learning in the adaptive controller. Long-term learning here is characterized by better approximation of the uncertainty over the entire domain. Long term learning results due to the convergence of the NN weights to a neighborhood of their ideal values (in sense of the Universal Approximation Property stated in Theorem 5.2.1). Figure 29 presents an interesting comparison between the online NN outputs ($W^T(t)\sigma(\bar{x}(t))$) for all the controllers, and the NN output with the weights and centers *frozen* at their values at the final time T of the simulation ($W^T(T)\sigma(\bar{x}(t))$), then re-simulated with these "learned weights". Figure 29a compares the online adaptation performance during the simulation, that is it compares $\Delta(\bar{x}(t))$ with $W^T(t)\sigma(\bar{x}(t))$. We can see that BKR-CL does the best job approximating the uncertainty online, particularly in the transient, whereas the classical laws tend to perform similar to each other. In Figure 29b, we see that the difference is more dramatic when the

weights are frozen at their learned values during the simulation; only BKR-CL completely captures the transient, and approximates the steady-state uncertainty fairly closely, while the classical laws do quite poorly. This implies that the weights learned with BKR-CL have converged to the values that provide a good approximation of the uncertainty over the entire operating domain. On the other hand, in Figure 29b we see that if BKR-CL is not used, the NN does not approximate the uncertainty as well, indicating that the weights have not approached their ideal values during the simulation. From Figure 25 we see that since the weights for e-mod and the projection operator are closer to those approached by BKR-CL than σ-mod, and they do slightly better in uncertainty approximation. The effect of long-term learning can be further characterized by studying the behavior of the BKR algorithm without concurrent learning and with a higher learning rate ($\Gamma_W = 10$). Higher learning rates are often used in traditional adaptive control (without concurrent learning) to guarantee good tracking. Figure 30a compares the online adaptation performance. Except for the initial transient, the NN outputs of the classical laws approximate the uncertainty well locally in time with a higher learning rate. However, Figure 30b shows that they do a poor job in estimating the ideal weights with the learning rate increased, indicating that good local approximation of the uncertainty through high adaptation gains does not necessarily guarantee long-term learning. This is motivated by the fact that the classical laws relying on the gradient based weight update law of (5.3.5) drive the weights only in the direction of maximum reduction of *instantaneous* tracking error cost, and hence, do not guarantee that the weights are driven close to the ideal values that best represent the uncertainty globally. On the other hand, BKR-CL has a better long-term adaptation performance as not only are the centers updated to best reflect the domain of the uncertainty but the weights are also driven towards the ideal values that best represent the uncertainty over the domain visited.

Table 6: Wingrock computation time comparison with non-BKR controllers.

Centers	e**-mod**	σ**-mod**	**proj**	**BKR-CL**
8	3.6	3.1	3.2	13.9
10	3.7	3.2	3.3	16.1
12	3.8	3.4	3.5	18.9
14	3.8	3.4	3.6	23.8

Table 7: Wingrock computation time comparison with BKR controllers.

Centers	**BKR** e**-mod**	**BKR proj**	**BKR** σ**-mod**	**BKR-CL**
8	4.2	4.2	4.4	13.9
10	4.4	4.4	4.5	16.1
12	4.6	4.6	4.7	18.9
14	4.8	4.8	5.0	23.8

5.6.2 Computational Complexity

The accuracy of the algorithm comes at a computational cost. The two main components of BKR-CL are the BKR step (Algorithm 11) and the CL step (Algorithm 12). Once the budget of the dictionary N_D is reached, the BKR step roughly takes $O(N_D^2)$ steps to add a new center, and is thus reasonably efficient. The main computational bottleneck in BKR-Cl is the computation of SVD of the history stack matrix Ω in the CL step. If there are p points in Ω, this takes $O(p^3)$ steps. Given the fact that p is usually larger than N_D, this can be quite expensive. Table 6 shows a computation time comparison between the non-BKR controllers and BKR-CL for the Wingrock simulation run for 60 seconds in MATLAB on a dual core Intel 2.93 Ghz processor with 8 GB RAM, while Table **??** shows the same comparison between BKR controllers and BKR-CL. As can be seen, the computation time increases dramatically for BKR-CL as the size of Ω increases. There are two things to note here however; the number of points added to the history stack Ω is determined by the tolerance ϵ in Algorithm 12. Therefore, if too many points are being added to the history stack, one can increase the tolerance, or set a separate budget to make p fixed. Secondly, we

used the native implementation of the SVD in MATLAB for these experiments, which is not optimized for an online setting. Recent work in incremental SVD computation (see for example [9, 10, 86]) speeds up the decomposition considerably, resulting in $\mathcal{O}(p^2)$ complexity once the first SVD with p points has been computed. Therefore, the computation time for the SVD component of BKR-CL can be made roughly comparable to the BKR component, which implies that the algorithm will scale better as the number of points in the history stack increases.

5.7 Conclusion

In this work, we established a connection between kernel methods and Persistently Exciting (PE) signals using RKHS theory. Particularly, we showed that in order to ensure PE, not only do the system states have to PE, but the centers need to be also selected in such a way that the mapping from the state space to the underlying RKHS is never orthogonal to the linear subspace generated by the Radial Basis Function (RBF) centers. This ensures that the output of the radial basis function does not vanish. We used this connection to motivate an algorithm called Budgeted Kernel Restructuring (BKR) that updates the RBF network in a way that ensures any inserted excitation is retained. This enabled us to design adaptive controllers without assuming any prior knowledge about the domain of the uncertainty. Furthermore, it was shown through simulation that on a budget (limitation on the maximum number of RBFs used), the method is better at capturing the uncertainty than evenly spaced centers, because the centers are selected along the path of the system through the state space. We augmented BKR with Concurrent Learning (CL), a method that concurrently uses specifically recorded and instantaneous data for adaptation, to create the BKR-CL adaptive control algorithm. It was shown through Lyapunov-like analysis that the weights for the centers picked by BKR-CL are bounded without needing PE. Simulations showed improved tracking performance.

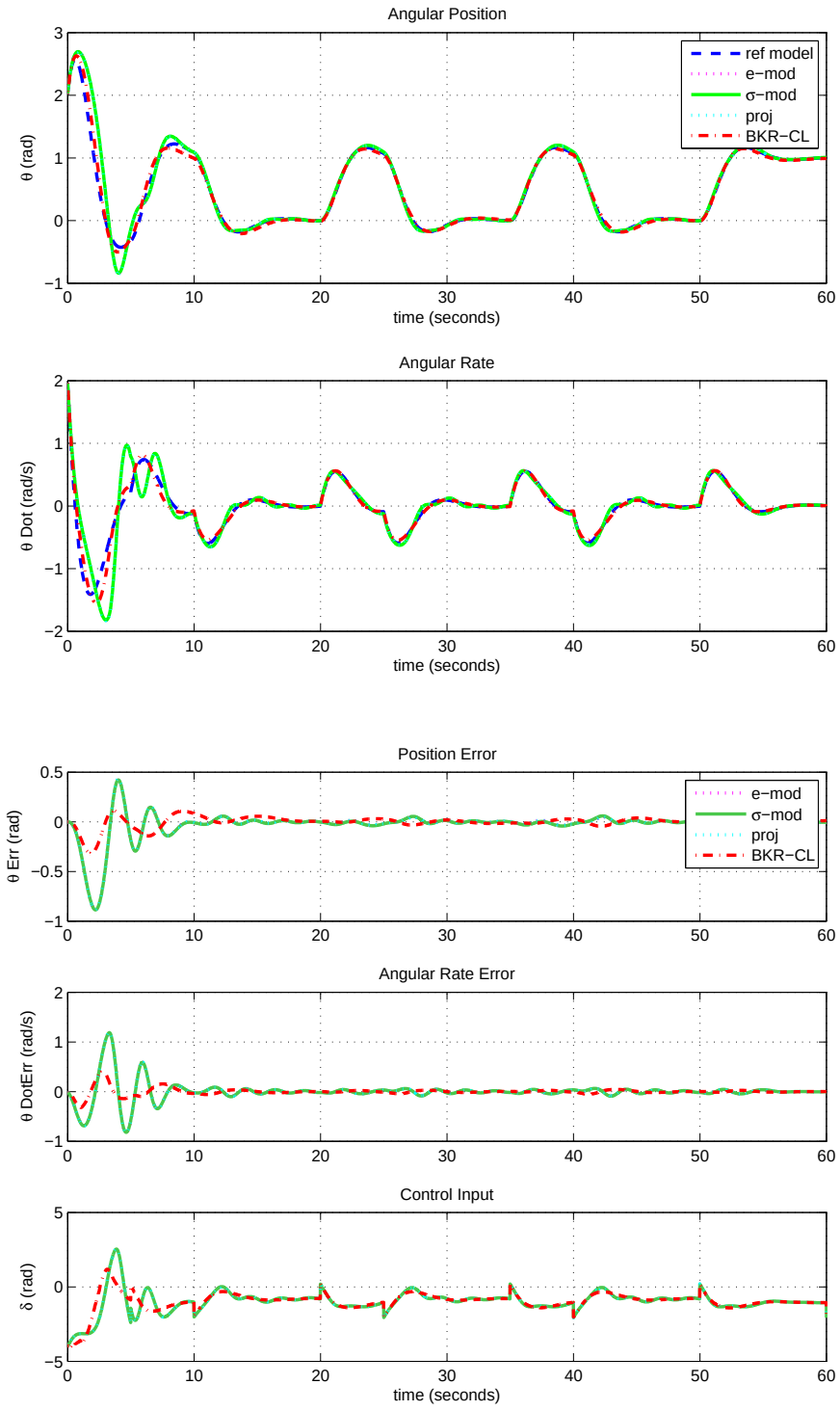


Figure 24: Tracking with e-mod, σ-mod, the projection operator, and BKR-CL.

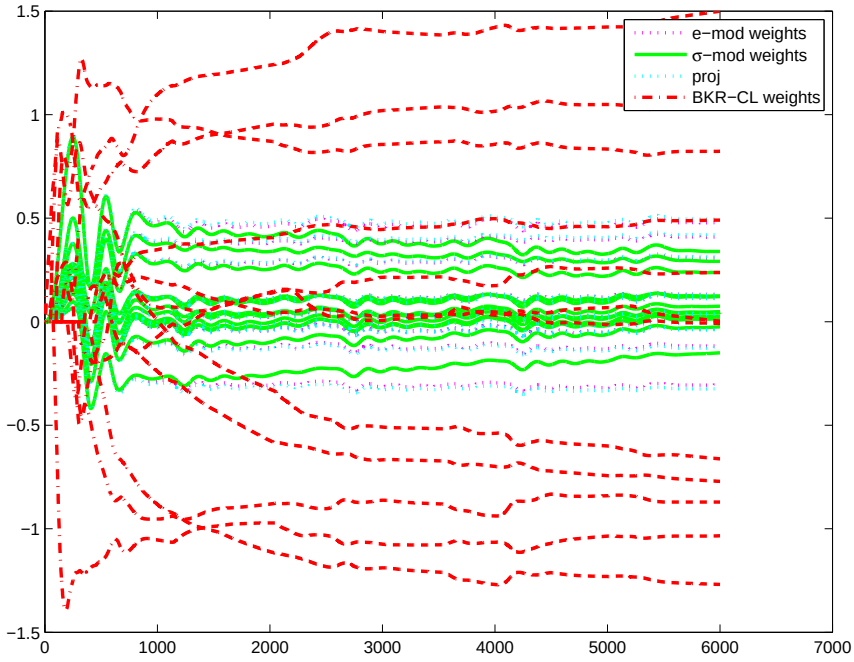


Figure 25: Weight evolution with e-mod, σ-mod, the projection operator, and BKR-CL.

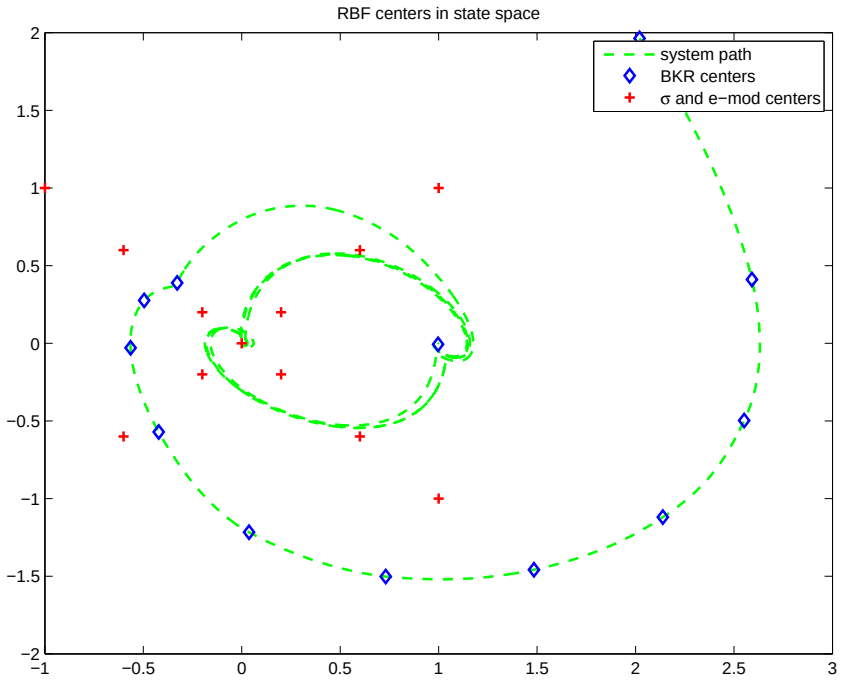


Figure 26: Center placement for the classical laws and BKR-CL.

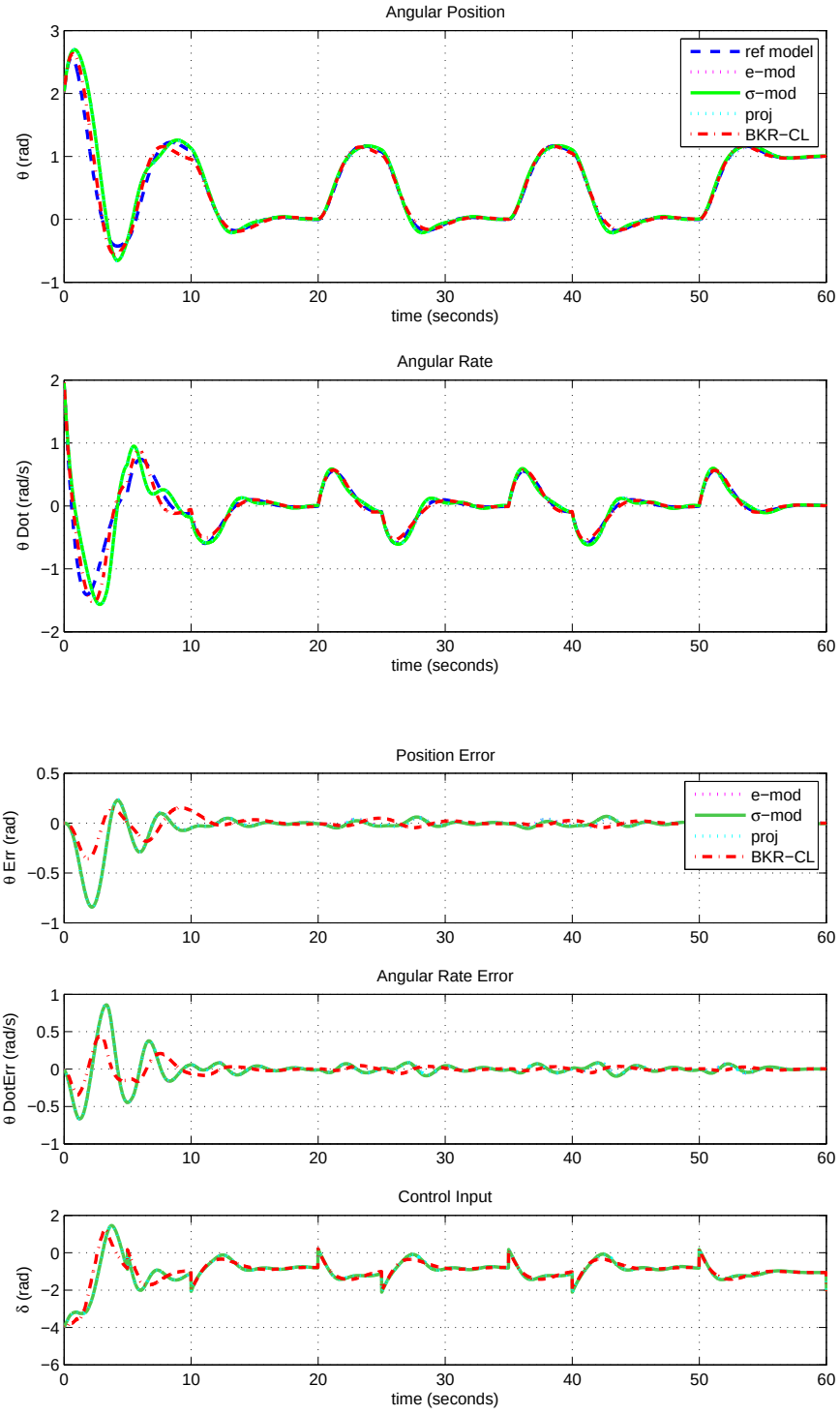


Figure 27: Tracking with BKR e-mod, BKR σ-mod, BRK proj and with BKR and concurrent learning (BKR-CL).

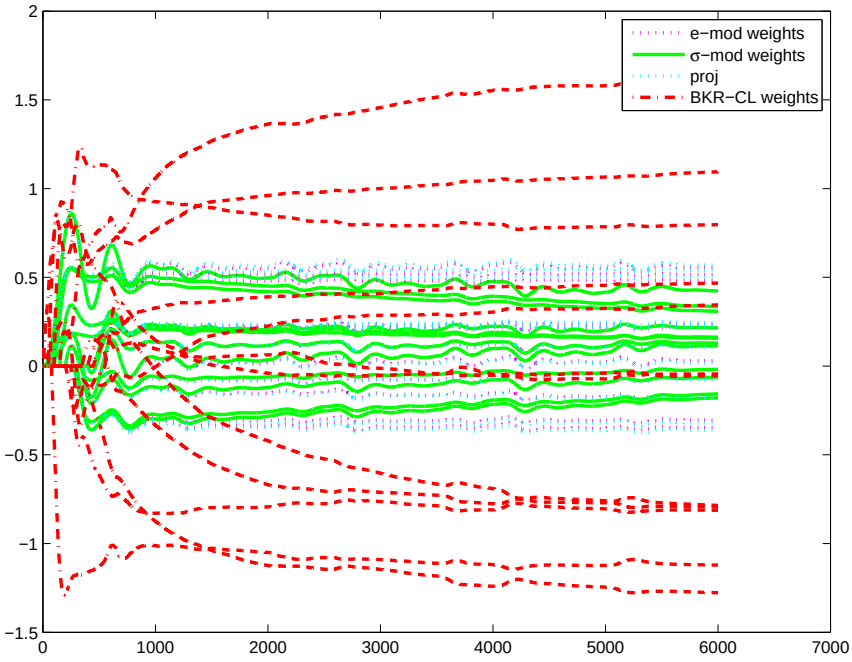
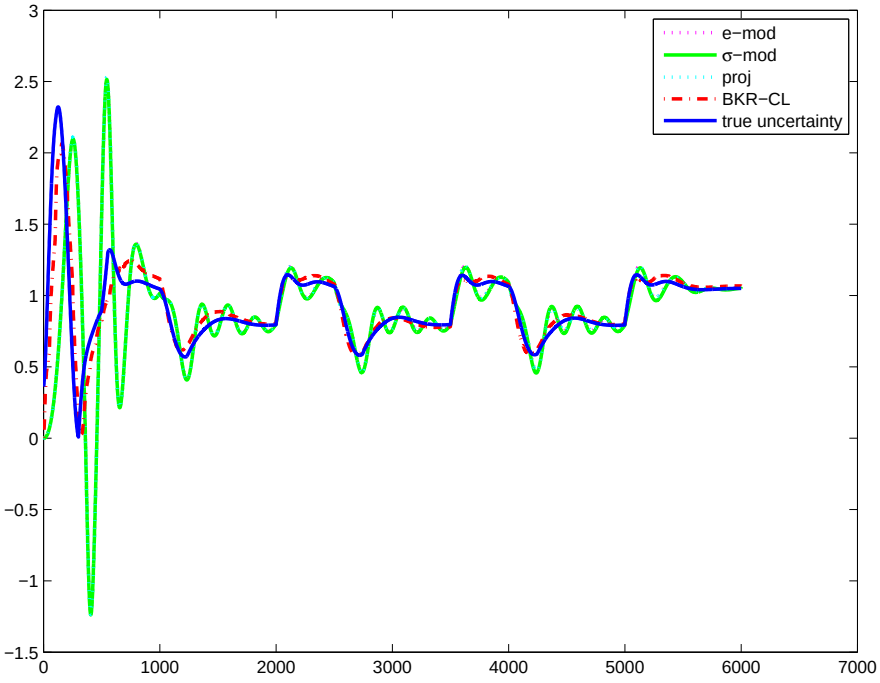
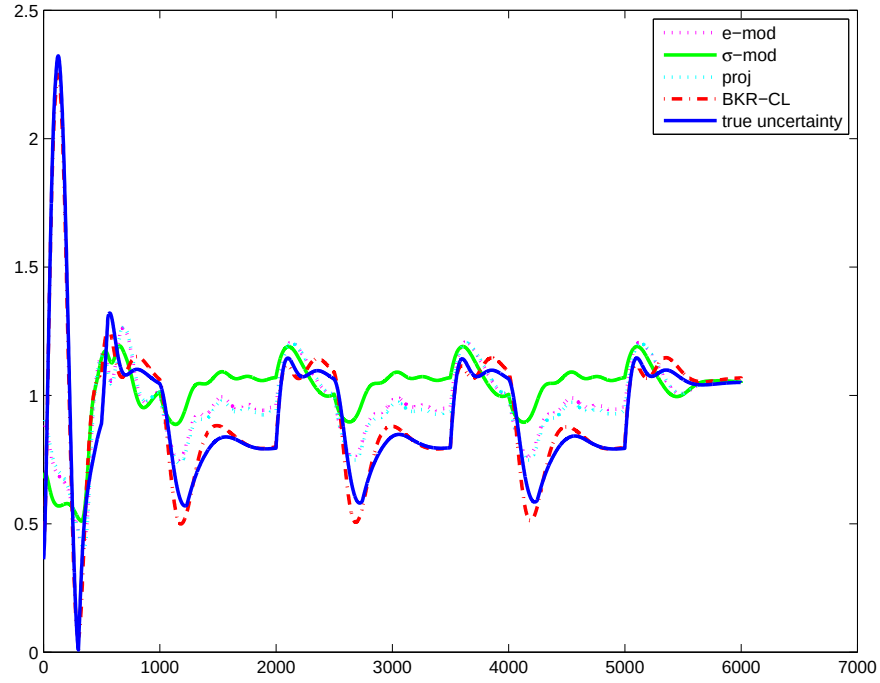


Figure 28: Weight evolution with BKR-CL and with BKR e-mod, BKR σ-mod and BKR proj.

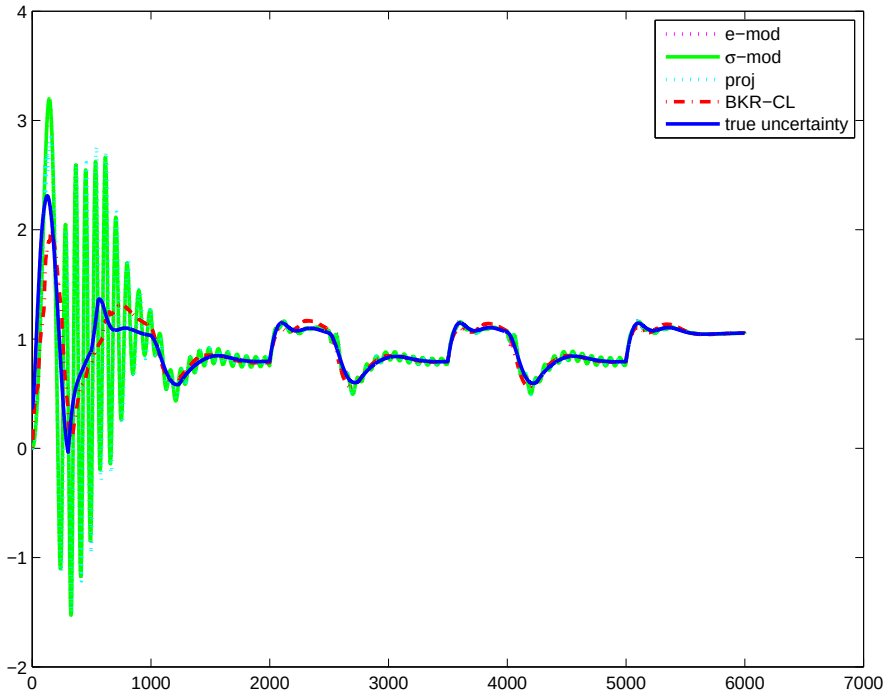


(a) Online uncertainty tracking

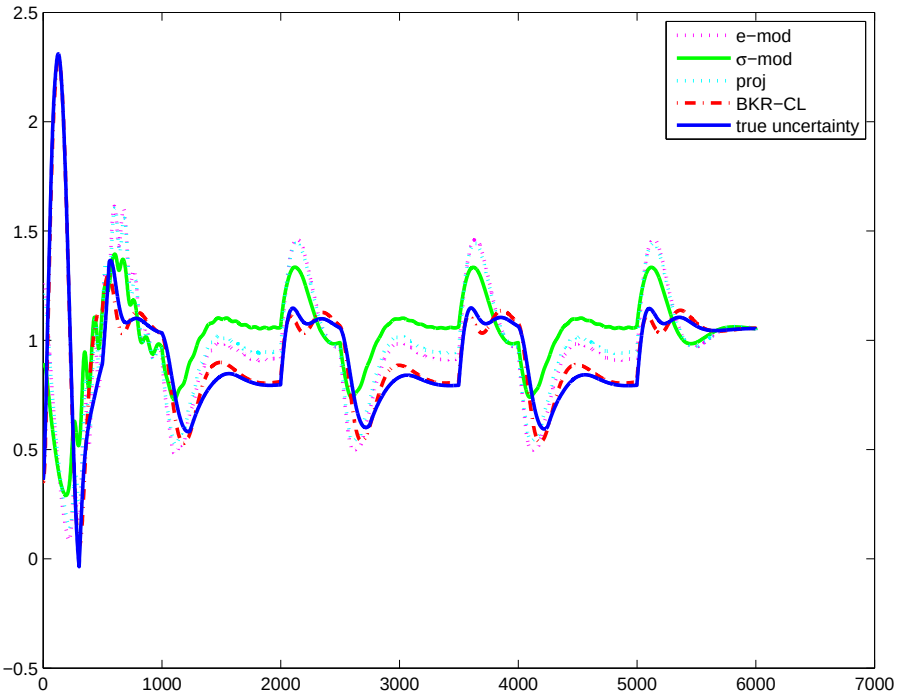


(b) Learned weight uncertainty tracking

Figure 29: Plots of the online NN output against the system uncertainty, and the NN output using the final weights frozen at the end of the simulation run. Note that the NN weights and centers with BKR-CL better approximate the uncertainty in both cases.



(a) Online uncertainty tracking



(b) Learned weight uncertainty tracking

Figure 30: Plots of the online NN output against the system uncertainty, and the NN output using the final weights frozen at the end of the simulation run, with a high learning rate.

CHAPTER VI

BAYESIAN NONPARAMETRIC ADAPTIVE CONTROL USING GAUSSIAN PROCESSES

As noted in the previous chapter, most current Model Reference Adaptive Control (MRAC) methods rely on parametric adaptive elements, in which the number of parameters of the adaptive element are fixed a priori, often through expert judgment. An example of such an adaptive element are Radial Basis Function Networks (RBFNs), with RBF centers pre-allocated based on the expected operating domain. If the system operates outside of the expected operating domain, this adaptive element can become non-effective in capturing and canceling the uncertainty, thus rendering the adaptive controller only semi-global in nature. This chapter investigates a Gaussian Process (GP) based Bayesian MRAC architecture (GP-MRAC), which leverages the power and flexibility of GP Bayesian nonparametric models of uncertainty. GP-MRAC does not require the centers to be preallocated, can inherently handle measurement noise, and enables MRAC to handle a broader set of uncertainties, including those that are defined as distributions over functions. We use stochastic stability arguments to show that GP-MRAC guarantees good closed loop performance with no prior domain knowledge of the uncertainty. Online implementable GP inference methods are compared in numerical simulations against RBFN-MRAC with preallocated centers and are shown to provide better tracking and improved long-term learning.

6.1 MOTIVATION

Within the class of MRAC algorithms, Radial Basis Function Neural Networks (RBFNs) have emerged as a widely used universal-approximator adaptive model [85], especially

when little is known about the uncertainty [46, 70, 102]. One reason for the popularity of RBFNs is that they are linear-in-the-parameters, as opposed to multi-layer perceptron Neural Networks [46, 56]. The accuracy of an RBFN representation, however, greatly depends on the choice of RBF centers [73]. Typically, authors have assumed that the operating domain of the system is known, and pre-allocated a fixed quantity of Gaussian RBF centers over the presumed domain [75, 108]. Therefore, an implicit assumption in RBFN-MRAC is that the domain over which the uncertainty is defined is compact, so that it can be well approximated by a finite set of RBFs. Consequently, RBFN stability results cannot be extended globally for if the system states were to evolve outside of the assumed compact domain of operation, the RBFN is effectively unable to learn and approximate the uncertainty. Another limitation of traditional RBFN based MRAC methods is that they model the uncertainty as a smooth deterministic function. However, in real-world situations uncertainties may have several stochastic effects, such as noise, servo chattering, turbulence etc. Authors have relied on σ-modification [37] like damping terms in parameter update equations for guaranteeing the boundedness of parameters of deterministic models of uncertainty in presence of noise [16, 51]. This approach can guarantee overall boundedness of the closed-loop system, but the added damping limits learning of the uncertainty. Therefore there is a need for better models of stochastic uncertainties using probabilistic modeling notions such as mean and variance.

This chapter outlines a method employing Gaussian Processes (GP) as Bayesian nonparametric adaptive elements in MRAC to address the aforementioned limitations of RBFN-MRAC with preallocated centers. As an alternative to methods with fixed parametric structure, nonparametric models are designed to overcome the local approximation properties of universal approximators. The number of parameters and their properties are not fixed, rather, they grow and adjust with the data. Within the class of nonparametric modeling methods, Bayesian modeling approaches lead

to data-driven, generative models, where the model is optimized to fit the data assuming uncertain measurements. GP based models are an example of a Bayesian nonparametric regression model [78]. GPs are known to have a deep connection with kernel filtering methods through a reproducing kernel Hilbert space interpretation of GP regression [2]. The benefits of using GP Bayesian nonparametric adaptive elements include: no prior knowledge of the operating domain of the uncertainty is required, measurement noise is inherently handled, and the centers need not be pre-allocated. GP uncertainties are defined via *distributions over functions*, which differs from the traditional deterministic weight-space based approaches [46, 47, 70, 108]. Furthermore, Bayesian inference overcomes the shortcomings of the standard gradient based MRAC parameter update laws, such as the lack of convergence guarantees and the possibility of bursting (parameters growing unboundedly) in presence of noise [3, 8, 68].

When used in an online setting such as for adaptive control, the number of parameters in GP adaptive elements grows linearly with the measurements [78]. Furthermore, most sparsification techniques used to limit the size of the GP model require access to the entire set of training data [57, 78, 90], which cannot be provided in real-time control settings. Therefore, in order to ensure real-time feasibility, we enforce an additional restriction that the number of maximum allowable parameters at any instant of time be limited (this number is referred to as the "budget"). Once the budget is reached, any new centers are added by removing older (possibly irrelevant) centers. Each time a new center is added, the structure of the GP changes discontinuously. Thus, the stability properties of the presented GP-MRAC controllers must be established through switched stochastic stability theory. Our GP-MRAC approach removes long-standing assumptions on bounded domain of operation, a priori known number and location of RBF centers, and deterministic input-output models of the uncertainty, while being on-line implementable.

6.1.1 Related Work

Nonparametric models in adaptive control have been previously considered. Cannon and Slotine presented a heuristic algorithm to add and remove nodes of a nonparametric wavelet network over a bounded domain for adaptive control [14]. Bernard and Slotine established stable adaptive controllers using finite approximations of infinite series of wavelet function regressors [6]. In contrast to the nonparameteric models in those aforementioned works, our approach leverages information theoretic concepts such as Bayesian inference and Kullback-Leibler divergence for updating the weights and selecting the regressors [32]. Furthermore, our approach does not assume a predefined bounded domain of operation. Alpcan investigated active optimal control with Gaussian processes in the context of dual control [1]. Murray-Smith *et al* have explored Gaussian processes in the context of dual adaptive control [63, 64]. Nguyen-Tuong and Peters combined a physics based model with a Gaussian Process model for learning inverse dynamics of stable systems such as robotic arms [72]. Ko and Fox explored GP based dynamic observation and measurement models in Kalman filter frameworks [49]. Diesenroth, Ko, Rasmussen, and others have used GPs for solving control problems using model based reinforcement learning [25, 48, 77]. The above approaches require access to off-line training data for generating control models. The key difference between these methods and our approach is that GPs are used here in the MRAC framework, which is a direct adaptive control framework (output of the online trained GP is directly used for control). Similar to traditional MRAC, the focus of our architecture is to guarantee stability and good tracking of fast, unstable dynamical systems online the *first-time-around*, without using any previously recorded data. This objective is achieved by leveraging recent advances in sparse online GP learning [24] and providing stability guarantees using a new analysis approach that utilizes stochastic control theory for switched uncertain systems.

In the previous chapter, we took the first steps to unite recent advances in both

kernel methods and adaptive control. The adaptive control system used non-Bayesian nonparametric kernel models with gradient based update laws. The work in this chapter incorporates Bayesian nonparametric regression techniques in adaptive control. Preliminary ideas regarding the work presented here first appeared in the conference paper [19]. The main contributions of the current work are improved techniques for online sparsification and inference, detailed stability analysis, and more complete analysis of numerical simulations.

6.1.2 Notation

Let $\mathbb{E}$ denote the expectation operator, $\mathbb{E}_x$ the expectation operator conditioned on the measurement x, and $\mathbb{V}(x)$ the variance of x. The trace operator is denoted by tr. The class of n^{th} continuously differentiable functions is denoted by $\mathbb{C}^n$. The operators $\lambda_{\min}(.)$ and $\lambda_{\max}(.)$ return the minimum and the maximum eigenvalue of a matrix. For a matrix Z, $|Z|$ denotes the number of columns. Lastly, $t \geq 0$ denotes time, where the argument t is sometimes dropped from an equation for ease of exposition.

6.2 Approximate Model Inversion based Model Reference Adaptive Control (AMI-MRAC)

AMI-MRAC is an approximate feedback-linearization based MRAC method that allows the design of adaptive controllers for a general class of nonlinear plants (see e.g. [12, 39]). The GP-MRAC approach is introduced in the framework of AMI-MRAC; unlike §5.3, the treatment presented here is a little bit more restrictive in that it focuses on multiple-input controllable control-affine nonlinear uncertain dynamical systems. Let $x(t) = [x_1^T(t), x_2^T(t)]^T \in D_x \subset \mathbb{R}^n$, such that $x_1(t) \in \mathbb{R}^{n_s}$, $x_2(t) \in \mathbb{R}^{n_s}$, and $n = 2n_s$. Let $u \in D_u \subset \mathbb{R}^l$, and consider the system

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= f(x(t)) + b(x(t))u(t). \end{aligned} \tag{6.2.1}$$

The functions $f(t)$, $f(0) = 0$ and b are partially unknown functions assumed to be Lipschitz over a domain $\mathcal{D}$ and the control input u is assumed to be bounded and piecewise continuous, so as to ensure the existence and uniqueness of the solution to (6.2.1) over $\mathcal{D}$. Also assume that $l \leq n_s$ (while restrictive for overactuated systems, this assumption can be relaxed through the design of appropriate control assignment [29]).

The AMI-MRAC approach used here feedback linearizes the system by finding a pseudo-control input $\nu(t) \in \mathbb{R}^{n_s}$ that achieves a desired acceleration. If the exact plant model in (6.2.1) is known and invertible, the required control input to achieve the desired acceleration is computable by inverting the plant dynamics. However, since this usually is not the case, an approximate inversion model $\hat{f}(x)+\hat{b}(x)u$, where $\hat{b}$ chosen to be nonsingular for all $x(t) \in D_x$, is employed.

Given a desired pseudo-control input $\nu \in \mathbb{R}^{n_s}$ a control command u can be found by approximate dynamic inversion:

$$u = \hat{b}^{-1}(x)(\nu - \hat{f}(x)). \tag{6.2.2}$$

Let $\bar{x} = (x^T, u^T)^T \in \mathbb{R}^{n+l}$ for brevity. The use of an approximate model leads to modeling error Δ for the system,

$$\dot{x}_2 = \nu(\bar{x}) + \Delta(\bar{x}), \tag{6.2.3}$$

with

$$\Delta(\bar{x}) = f(x) - \hat{f}(x) + (b(x) - \hat{b}(x))u. \tag{6.2.4}$$

Were b known and invertible with respect to u, then an inversion model exists such that the modeling error is not dependent on the control input u. A designer chosen reference model is used to characterize the desired response of the system

$$\begin{aligned} \dot{x}_{1_{rm}} &= x_{2_{rm}}, \\ \dot{x}_{2_{rm}} &= f_{rm}(x_{rm}, r), \end{aligned} \tag{6.2.5}$$

where $x_{i_{rm}}$ and $f_{rm}(x_{rm}, r)$ denote the reference model states and dynamics respectively, and $f_{rm}(x_{rm}, r)$ is assumed to be continuously differentiable in x_{rm} for all $x_{rm} \in D_x \subset \mathbb{R}^n$. The command $r(t)$ is assumed to be bounded and piecewise continuous. Furthermore, f_{rm} is assumed to be such that x_{rm} is bounded for a bounded command.

Define the tracking error to be $e(t) = x_{rm}(t) - x(t)$, and the pseudo-control input ν to be

$$\nu = \nu_{rm} + \nu_{pd} - \nu_{ad}, \tag{6.2.6}$$

consisting of a linear feedforward term $\nu_{rm} = \dot{x}_{2_{rm}}$, a linear feedback term $\nu_{pd} = [K_1, K_2]e$ with $K_1 \in \mathbb{R}^{n_s \times n_s}$ and $K_2 \in \mathbb{R}^{n_s \times n_s}$, and an adaptive term $\nu_{ad}(\bar{x})$. For ν_{ad} to be able to cancel Δ, the following assumption needs to be satisfied:

Assumption 6.2.1. *There exists a unique fixed-point solution to* $\nu_{ad} = \Delta(x, \nu_{ad})$, $\forall x \in D_x$.

Assumption 6.2.1 implicitly requires the sign of control effectiveness to be known [45]. Sufficient conditions for satisfying this assumption are available in [45, 118].

Using (6.2.3) the tracking error dynamics can be written as

$$\dot{e} = \dot{x}_{rm} - \begin{bmatrix} x_2 \\ \nu + \Delta \end{bmatrix}. \tag{6.2.7}$$

Let $A = \begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$, where $0 \in \mathbb{R}^{n_s \times n_s}$ and $I \in \mathbb{R}^{n_s \times n_s}$ are the zero and identity matrices, respectively. From (6.2.6), the tracking error dynamics are then,

$$\dot{e} = Ae + B[\nu_{ad}(\bar{x}) - \Delta(\bar{x})]. \tag{6.2.8}$$

The baseline full state feedback controller ν_{pd} is chosen to make A Hurwitz. Hence, for any positive definite matrix $Q \in \mathbb{R}^{n \times n}$, a positive definite solution $P \in \mathbb{R}^{n \times n}$ exists

for the Lyapunov equation

$$0 = A^T P + PA + Q. \tag{6.2.9}$$

When Gaussian RBFN are used as adaptive elements, the adaptive part of the control law (6.2.6) is represented by a linear combination of Gaussian RBFs $\nu_{ad}(\bar{x}) = W^T \Phi(\bar{x})$ where $W \in \mathbb{R}^{n_2 \times q}$ and $\Phi(\bar{x}) = [1, \phi_2(\bar{x}), \phi_3(\bar{x}), ..., \phi_q(\bar{x})]^T$ is a q dimensional vector of radial basis functions.

6.3 ADAPTIVE CONTROL USING GAUSSIAN PROCESS REGRESSION

Traditionally MRAC assumes that the uncertainty, $\Delta(\bar{x})$ in (6.2.4), is a (smooth) deterministic function for which an input-output map in the form of an NN or RBFN is learned. This chapter offers an alternate view by assuming that the uncertainty is described by a time varying (prior) mean and covariance function, and using Gaussian Processes (GP) to learn the continuous function of time and state [78]. As the system evolves and measurements are taken, Bayesian posterior updates build a generative model of the uncertainty. Learning via probabilistic generative models offers a promising new approach. In contrast to learning a NN input-output representation [62], the approach does not easily succumb to overlearning and also handles noise. Furthermore, it allows incorporation of stochastic effects such as servo chattering, external disturbances, and other non-deterministic effects.

A GP is defined as a collection of random variables such that every finite subset is jointly Gaussian. The joint Gaussian condition means that GPs are completely characterized by their second order statistics [78]. A GP is a distribution over functions, that is, a draw from a GP is a function. For the sake of clarity of exposition, we will assume that $\Delta(\bar{x}) \in \mathbb{R}$; the extension to the multidimensional case is straightforward. When Δ follows a Gaussian process model, then

$$\Delta(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)), \tag{6.3.1}$$

where $m(\cdot)$ is the mean function, and $k(\cdot,\cdot)$ is a real-valued, positive definite covariance kernel function. Under GP regression, the mean is assumed to lie in the class of functions

$$\mathcal{G} = \left\{ g(\cdot) \in \mathbb{R}^{\mathcal{X}} \;\middle|\; g(\cdot) = \sum_{i=1}^{\infty} \alpha_i k(\bar{x}_i, \cdot) \right\}, \tag{6.3.2}$$

where $\mathcal{X} \subset \mathbb{R}^n$, $\alpha_i \in \mathbb{R}$, $\bar{x}_i \in \mathcal{X}$. The space $\mathcal{G}$ is a subspace of $\mathcal{H}$, a reproducing kernel Hilbert space (RKHS), and $\|g\|_{\mathcal{H}} < \infty$ where $\|g(\cdot)\|^2_{\mathcal{H}} = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \alpha_i \alpha_j k(\bar{x}_i, \bar{x}_j)$. This assumption imposes a smoothness prior on the mean and renders the problem more amenable to analysis though the representer theorem [88]. Figure 31 demonstrates the ability of GP regression to mitigate noise in the data. An accurate model of the underlying function is learned although the data is corrupted by Gaussian white noise of variance ω^2.

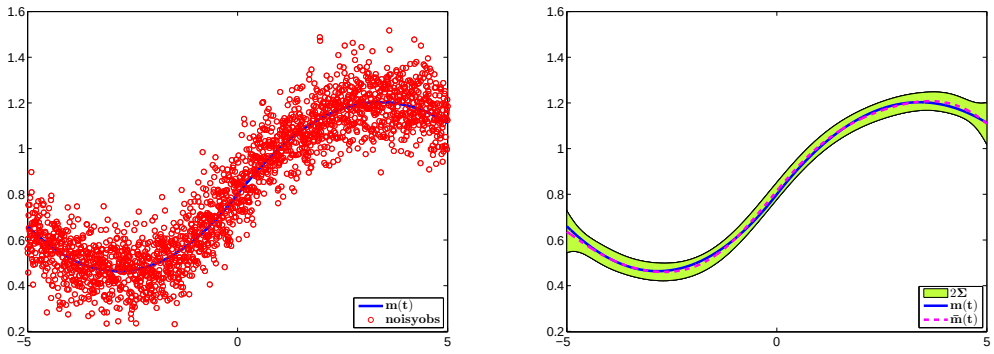


Figure 31: An example of GP inference for a set of $2,000$ measurements drawn from $\Delta(\bar{x}) \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$, corrupted by i.i.d observation noise drawn from $\mathcal{N}(0, \omega^2)$. $m(\cdot)$ is the estimated mean, and $\Sigma(\cdot)$ is the estimated posterior variance.

6.3.1 GP Regression and Reproducing Kernel Hilbert Spaces

Let $Z_\tau = \{\bar{x}_1, \ldots, \bar{x}_\tau\}$ be a set of state measurements, discretely sampled where $\{1 \ldots \tau\}$ are indices for the discrete sample times $\{t_1, \ldots, t_\tau\}$. The set defines a covariance matrix $K_{ij} := k(\bar{x}_i, \bar{x}_j)$. Given indexed sets A and B, $K(A, B)$ denotes the kernel matrix generated by the evaluations $K_{ij} = k(a_i, b_j)$ between the two sets, where $a_i \in A, b_j \in B$. For each measurement $\bar{x}_i$, there is an observed output $y(\bar{x}_i) =$

$m(\bar{x}_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \omega^2)$. The stacked outputs give $y = [y_1, \ldots, y_t]^T$. The most common choice of covariance kernel, and the one used here, is the Gaussian RBF kernel.

GP regression fuses RKHS theory with Bayesian linear regression by utilizing a regression model of the form

$$m(\bar{x}) = \beta^T \Psi(\bar{x}) = \sum_{i \in \mathcal{I}} \beta_i \langle \psi(\bar{x}_i), \psi(\bar{x}) \rangle_{\mathcal{H}}, \tag{6.3.3}$$

where $\beta \in \mathcal{F}_Z$ is a coordinate vector (of weights), $\psi(\bar{x}_i) \in \mathcal{H}$ are basis vectors, and $\mathcal{I}$ is the index set. Since GP adaptive elements admit infinite basis functions and associated parameters, they are referred to as nonparametric [78].

GP regression assumes that the uncertainty in the data and the model follow Gaussian distributions, while modeling the function estimate using a mean function $\hat{m}$ and a covariance function $\hat{\Sigma}$. Since the observations are Gaussian, the likelihood function $p(y_\tau | Z_\tau, \beta)$ is Gaussian. The initial prior is set to $p(\beta) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_w})$, and Bayes' rule is used to infer the posterior distribution $p(\beta | Z_\tau, y_\tau)$ with each new observation. Since the posterior is Gaussian, the update generates a revised mean $\hat{m}_\tau$ and covariance $\hat{\Sigma}_\tau$. If $|Z_\tau|$ is finite, the solution for the posterior mean and covariance is also finite [88]. In particular, given a new input $\bar{x}_{\tau+1}$, the joint distribution of the data available up to τ and $\bar{x}_\tau$ under the prior distribution is

$$\begin{bmatrix} y_\tau \\ y_{\tau+1} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(Z_\tau, Z_\tau) + \omega^2 I & k_{\bar{x}_{\tau+1}} \\ k_{\bar{x}_{\tau+1}}^T & k_{\tau+1}^* \end{bmatrix}\right), \tag{6.3.4}$$

where $k_{\bar{x}_{\tau+1}} = K(\bar{x}_{\tau+1}, Z_\tau)$ and $k_{\tau+1}^* = k(\bar{x}_{\tau+1}, \bar{x}_{\tau+1})$. The posterior (sometimes called the predictive) distribution, obtained by conditioning the joint Gaussian prior distribution over the observation $\bar{x}_{t+1}$, is computed by

$$p(y_{\tau+1} | Z_\tau, y_\tau, \bar{x}_{\tau+1}) \sim \mathcal{N}(\hat{m}_{\tau+1}, \hat{\Sigma}_{\tau+1}), \tag{6.3.5}$$

where

$$\hat{m}_{\tau+1} = \beta_{\tau+1}^T k_{\bar{x}_{\tau+1}} \tag{6.3.6}$$

$$\hat{\Sigma}_{\tau+1} = k_{\tau+1}^* - k_{\bar{x}_{\tau+1}}^T C_\tau k_{\bar{x}_{\tau+1}} \tag{6.3.7}$$

are the updated mean and covariance estimates, respectively, and where $C_\tau := (K(Z_\tau, Z_\tau) + \omega^2 I)^{-1}$ and $\beta_{\tau+1} := C_\tau y_\tau$. Due to the Representer Theorem, the values in (6.3.6) are the best possible that could be obtained given the available data [78, 88].

Since both Z_τ and y_τ grow with data, computing the inverse becomes computationally intractable over time. This is less of a problem for traditional GP regression applications, which often involve finite learning samples and offline learning. However, in an online setting, the linear growth in the sample set cardinality degrades computational performance. Therefore, the extension of GP regression to MRAC, requires an online method to restrict the number of data points stored for inference. In the following section we outline two simple schemes for this purpose and incorporate them with MRAC to form GP-MRAC.

6.3.2 GP Bayesian nonparametric model based MRAC

We model the uncertainty using a Gaussian process based adaptive element ν_{ad}

$$\nu_{ad}(\bar{x}) \sim \mathcal{GP}(\hat{m}(\bar{x}), k(\bar{x}, \bar{x}')), \tag{6.3.8}$$

where $\hat{m}(z)$ is the estimate of the mean of (6.3.1) updated using (6.3.6) with the coordinate vector denoted by α instead of β. The adaptive signal ν_{ad} is set to the estimate of the mean $\hat{m}(\bar{x})$. While the posterior calculated in Equation (6.3.5) converges to the true posterior [78], it becomes untenable in an online setting due to the growth of $|Z|$. There needs to be a limit on the number of datapoints stored for posterior inference.

Many schemes exist for the approximation of a GP with a smaller set of bases, but these typically assume all the data is available. The schemes, not designed for

Algorithm 13 The Generic Gaussian Process - Model Reference Adaptive Control (GP-MRAC) algorithm

Input: GP model parameters σ and ω^2, kernel approximation tolerance ϵ_{tol}, error tolerance ϵ_e.

Procedure:
while new measurements $\Delta(\bar{x}_{\tau+1})$ are available **do**
  Given $\bar{x}_{\tau+1}$, compute $\gamma_{\tau+1}$ using (5.4.1).
  Compute $y_{\tau+1} = \dot{\hat{x}}_{2_{\tau+1}} - \nu_{\tau+1}$.
  Compute tracking error norm $\|e(t)\|$.
  if $\gamma_{\tau+1} > \epsilon_{tol}$ or $\|e(t)\| > \epsilon_e$ **then**
    if $|\mathcal{BV}(\sigma)| > p_{\max}$ **then**
      Delete element in $\mathcal{BV}(\sigma)$.
    end if
    Add new basis vector based on methods in Section 6.3.2.
    Increase the switching index σ.
  end if
  Calculate $\hat{m}_{\tau+1}$ and $\hat{\Sigma}_{\tau+1}$.
  Set $\nu_{ad} = \hat{m}_{\tau+1}$.
  Calculate pseudo control ν using (6.2.6).
  Calculate control input using (6.2.2).
end while

an online setting, can be computationally costly [13, 33]. Since the set Z generates a family of functions $\mathcal{F}_Z \subset \mathcal{H}$ whose richness characterizes the quality of the posterior inference, a natural and simple way to determine whether to add a new point to the subspace is to check how well it is approximated by the elements in Z. To do this, we use the kernel linear independence test outlined in §5.4. When the budget is exceeded, a basis vector element must be removed prior to adding another element [13]. Here we examine two such schemes. The first, denoted **OP**, simply deletes the oldest vector: this scheme prioritizes temporal locality of the data for the approximation. The second, denoted **KL**, employs the *sparse online Gaussian process* algorithm. The latter algorithm efficiently approximates the KL divergence between the current GP and the $(t+1)$ alternative GPs missing one data point each, then deletes removes the data point with the largest KL divergence. See [24] and the appendix for this chapter for more details.

6.4 Analysis of Stability

6.4.1 Stochastic Stability Theory for Switched Systems

We begin by introducing the necessary tools in stochastic stability analysis. Consider switched stochastic differential equations of the Itô type whose solutions are a class of continuous time Markov processes [44, 52]. The system equations are

$$dx(t) = F(x(t))dt + G_\sigma(x(t))d\xi(t), \quad x(0) = x_0, \tag{6.4.1}$$

where $x \in \mathbb{R}^{n_s}$, $\xi(t) \in \mathbb{R}^{n_2}$ is a Wiener process, $\sigma(t) \in \mathbb{N}$ is the switching index which switches finitely many times in any finite time interval, $F(x(t))$ is an n_s-vector function, and G_σ is an $n_s \times n_2$ matrix. Assume that $F(0) = 0$ and $G_{\sigma(0)} = 0$. The functions $F(x(t))$ and G_σ are assumed to satisfy the Lipschitz condition for each switching index σ

$$\|F(x) - F(y)\| + \|G_\sigma(x) - G_\sigma(y)\| \leq B\|x - y\|$$

for all x and y. Under these conditions the solution of (6.4.1) is a continuous Markov process. Note that the assumption on Lipschitz continuity of G_σ is reasonable for the GP formulation considered here because components of G_σ turn out to be continuously differentiable kernel functions.

The following definitions concerning the (exponential) ultimate boundedness of the solution of (6.4.1) are introduced.

Definition 6.4.1. *The process $x(t)$ is said to be* mean square ultimately bounded uniformly in σ *if there exists a positive constant K such that for all t, $x_0 \in \mathbb{R}^{n_s}$, and σ,*

$$\lim_{t\to\infty} \mathbb{E}_{x_0}\|x(t)\|^2 \leq K. \tag{6.4.2}$$

Definition 6.4.2. *The process $x(t)$ is said to be* exponentially mean square ultimately bounded uniformly in σ *if there exist positive constants K, c, and α such that for all*

t, $x_0 \in \mathbb{R}^{n_s}$, and for all σ

$$\mathbb{E}_{x_0}\|x(t)\|^2 \leq K + c\|x_0\|^2 e^{-\alpha(t)}. \tag{6.4.3}$$

The Itô differential generator $\mathcal{L}$ for the smooth function $V(x(t))$ is given by

$$\mathcal{L}V(x(t)) = \frac{\partial V(x(t))}{\partial t} + \sum_j F_j(x(t))\frac{\partial V(x(t))}{\partial x_j} + \frac{1}{2}\sum_{i,j}\left[G_\sigma G_\sigma^T\right]_{ij}(x(t))\frac{\partial^2 V(x(t))}{\partial x_j x_i}. \tag{6.4.4}$$

Here, $\left[G_\sigma G_\sigma^T\right]_{ij}$ is the i^{th} row and j^{th} column element in the $n_s \times n_s$ matrix $G_\sigma G_\sigma^T$. The following lemma is a special case of that proved by Miyahara [60].

Lemma 6.4.1. *Let $x(t)$ be the process defined by the solution to (6.4.1). Let $V(x(t))$ be of class $\mathscr{C}^2$ with respect to x, of class $\mathscr{C}^1$ with respect to t, and bounded from below. If for some nonzero constants k_1, k_2, $\mathcal{L}V(x(t)) \leq k_1 - k_2 V(x(t))$ then $\mathbb{E}_{x_0}V(x(t)) \leq V(x_0)e^{-k_2 t} + \frac{|k_1|}{k_2}(1 - e^{-k_2 t})$ for all $t \geq 0$.*

The following theorem extends the ultimate boundedness results of Miyahara to switched Itô differential equation (6.4.1).

Theorem 6.4.1. *Let $x(t)$ be the process defined by the solution to (6.4.1), and let $V(x(t))$ be a function of class $\mathscr{C}^2$ with respect to x, and class $\mathscr{C}^1$ with respect to t. If,*

1. *$-\alpha_1 + c_1\|x\|^2 \leq V(x(t))$ for real α and $c_1 > 0$; and*
2. *$\mathcal{L}V(x(t)) \leq \beta_\sigma - c_2 V(x(t))$ where $\mathcal{L}$ is the differential generator of the Itô process as defined in (6.4.4), for real β_σ and $c_2 > 0$, and all switch states σ;*

then the process $x(t)$ is mean square ultimately bounded uniformly in σ. Suppose in addition $V(x(t)) \leq c_3\|x\|^2 + \alpha_2$, then the process $x(t)$ is exponentially mean square ultimately bounded uniformly in σ.

Proof. From Lemma 6.4.1,

$$\mathbb{E}_{x_0}V(x(t)) \leq V(x_0)e^{-c_2 t} + \frac{|\beta_\sigma|}{c_2}(1 - e^{-c_2 t}). \tag{6.4.5}$$

Therefore, $\lim_{t\to\infty}(\mathbb{E}_{x_0}V(x(t))) \to \frac{|\beta_\sigma|}{c_2}$. Since $-\alpha_1 + c_1\|x\|^2 \leq V(x(t))$, we have $\|x\|^2 \leq \frac{V(x(t))}{c_1} + \frac{\alpha_1}{c_1}$. Therefore it follows that

$$\mathbb{E}_{x_0}\|x(t)\|^2 \leq \frac{1}{c_1}\mathbb{E}_{x_0}V(x(t)) + \frac{\alpha_1}{c_1} \to \frac{|\beta_\sigma|}{c_1c_2} + \frac{\alpha_1}{c_1} \tag{6.4.6}$$

as $t \to \infty$. From which it follows that $\lim_{t\to\infty}\mathbb{E}_{x_0}\|x(t)\|^2 \leq K$ with $K = \max_\sigma(\frac{|\beta_\sigma|}{c_1c_2} + \frac{\alpha_1}{c_1})$. Therefore the process $x(t)$ is mean square ultimately bounded uniformly in σ. If in addition $V(x_0) \leq c_3\|x_0\|^2 + \alpha_2$, then from (6.4.5) and (6.4.6), $\mathbb{E}_{x_0}\|x(t)\|^2 \leq \frac{c_3}{c_1}\|x_0\|^2e^{-c_2t} + \frac{\alpha_2}{c_1} + K$. Hence, the process $x(t)$ is exponentially mean square ultimately bounded uniformly in σ. □

6.4.2 Analysis of Stability for GP-MRAC

This section establishes the stability of Gaussian process based MRAC using Algorithm 13. Let $\sigma(t) \in \mathbb{N}$ denote a switching index that increments every time the set $\mathcal{BV}$ is modified. When the σ^{th} system is active, the mean function estimate evaluated using the current set $\mathcal{BV}(\sigma)$, is denoted $\hat{m}^\sigma$. The results presented assume that the uncertainty can be expressed as

$$\Delta(\bar{x}) \sim m(\bar{x}(t)) + G_\sigma(\bar{x}(t))d\xi(t), \tag{6.4.7}$$

where $\xi(t)$ is a (zero-mean) Wiener process, and G_σ is a linear operator associated to the covariance kernel $k(\bar{x}, \bar{x}')$ (see Appendix 2 of [78]). Then, for a given σ, the tracking error dynamics of (6.2.8) are

$$de = Ae\,dt + B\left(\epsilon_m^\sigma(\bar{x})\,dt - G_\sigma(\bar{x}(t))\,d\xi\right), \tag{6.4.8}$$

where $\nu_{ad}(\bar{x}) \sim \mathcal{GP}(\hat{m}^\sigma(\bar{x}), k(\bar{x}, \bar{x}'))$, and $\epsilon_m^\sigma(\bar{x}) = \hat{m}^\sigma(\bar{x}) - m(\bar{x})$. For the sake of brevity, we drop the time and state dependency of $G_\sigma(\bar{x}(t))$ in the remaining section. First, note that $\{G_\sigma\,d\xi(t)\}_{t\in T}$ is a zero-mean Gaussian process.

We will need the following theorem.

Theorem 6.4.2. *(**Dudley** [28]) Define a pseudometric*

$$d_G(t,s) = \sqrt{\mathbb{E}[|G_\sigma\, d\xi(t) - G_\sigma\, d\xi(s)|]^2} \tag{6.4.9}$$

on T. Let $N(T,d,\upsilon)$ be the υ-covering number of the space; then the υ-entropy of the space (T,d) is given by $H(T,d,\upsilon) = \log N(T,d,\upsilon)$. Let $D(T)$ be the diameter of the space T with respect to the metric d_G. Then the following bound holds:

$$\mathbb{E}\sup_{t\in T} G_\sigma\, d\xi(t) \leq C\int_0^{D(T)} H^{1/2}(T,d,\upsilon)d\upsilon. \tag{6.4.10}$$

Corollary 6.4.1. *Let the covariance kernel $k : \Omega\times\Omega \to \mathbb{R}$ of the zero mean Gaussian process $\{G_\sigma\, d\xi(t)\}_{t\in T}$ be Gaussian. Then if $\Omega \subset \mathbb{R}^d$ is compact, there exists a constant $c' \in \mathbb{R}_+$ such that $\|G_\sigma\, d\xi(t)\| \leq c'$ almost surely.*

Proof. The RHS of (6.4.10) is the *Dudley entropy integral*; if it is finite, then the process $G_\sigma\, d\xi(t)$ is almost surely bounded [50]. Using $k(t,t') = \mathbb{E}[G_\sigma\, d\xi(t)G_\sigma\, d\xi(t')]$, we can write the above psuedometric as

$$\begin{aligned} d_G(t,s) &= \sqrt{\mathbb{E}[G_\sigma\, d\xi(t)G_\sigma\, d\xi(t)] + \mathbb{E}[G_\sigma\, d\xi(s)G_\sigma\, d\xi(s)] - 2\mathbb{E}[G_\sigma\, d\xi(t)G_\sigma\, d\xi(s)]} \\ &= \sqrt{k(t,t) + k(s,s) - 2k(t,s)}. \end{aligned}$$

In the case of radially-symmetric kernels, the above can be simplified to

$$d_G(t,s) = \sqrt{2}\sqrt{\kappa - k(t,s)},$$

where κ is the maximum of the kernel. Since $d_G(t,s) \leq \sqrt{2\kappa}$ for any choice of $t,s \in \mathbb{R}^d$, the diameter of the space is bounded, and thus $D(T) = \sqrt{2\kappa}$. Therefore, if the evolution of the system is over a compact domain, from the estimate on the covering numbers of the Gaussian kernel given in [53], the integral (6.4.10) is finite. □

Our goal is to prove the boundedness of the tracking error $e(t)$. Using the explicit solution for linear differential equations [42] and (6.4.8), we have that

$$e(t) = e^{Ae(t)}x_0 + \int_0^T e^{A(T-\tau)}B\epsilon_m^\sigma(\bar{x}(t))\,dt - \int_0^T e^{A(T-\tau)}BG_\sigma(\bar{x}(t))\,d\xi(t)$$

The matrix A is Hurwitz, so we have a bound on the first term, and using the above corollary, we can estimate the third term. To get an estimate on the second term, we need to bound $\epsilon_m^\sigma(\bar{x}(t))$. To keep the notation simple, we swap $\bar{x}$ for x in the following computation.

Recall that in GP inference, given data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in \Omega$ and $y_i = m(x_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \omega^2)$, we have

$$m(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x),$$

where

$$\alpha = \left(K + \omega^2 I\right)^{-1} y,$$

where $K_{ij} := k(x_i, x_j)$ and $y \in \mathbb{R}^n$ is a vector representation of the y values. If we had another kernel matrix induced by a quantization operator $\vartheta : \{1, \ldots, n\} \to \{1, \ldots, m\}$, such that $x_i \mapsto c_{\vartheta i}$, where $c_{\vartheta i} \in \Omega$ are a set of centers, we then get an approximate solution

$$\hat{m}^\sigma(x) = \sum_{i=1}^{n} \widetilde{\alpha}_i k(c_{\vartheta(i)}, x),$$

where

$$\widetilde{\alpha} = \left(\widetilde{K} + \omega^2 I\right)^{-1} y,$$

and where $\widetilde{K}_{ij} := k(c_{\vartheta i}, c_{\vartheta j})$. We also assume that the kernel is normalized, i.e.

$$\int_\Omega \int_\Omega k(x, y) dx dy = C,$$

and for finite sets,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j) \leq C.$$

For the sake of simplicity, we normalize $k(x, y)$ explicitly by dividing by n in our calculations in the next theorem. We can now prove a bound on $\epsilon_m^\sigma(\bar{x}(t))$.

Theorem 6.4.3. *(**Global Approximation Theorem**) Let h and $\widetilde{h}$ be defined as above, let $\sup_{x' \in \Omega} k(x', x) \leq \kappa$, and let $\|y\|_\infty \leq M$. Then*

$$\|\epsilon_m^\sigma(\bar{x})\| \leq \frac{2\kappa^2 M\sqrt{k_{\max}}}{\omega^4} + \frac{\kappa k_{\max} M}{\omega^2}, \tag{6.4.11}$$

where $k_{\max} := \max_i \left\|\psi(\bar{x}_i) - \psi(c_{\vartheta(i)})\right\|_{\mathcal{H}}$ is the greatest kernel approximation error.

Proof. We follow the proof of Theorem 4.2.1. Let $K_{ij} := n^{-2}k(x_i, x_j)$ and $\widetilde{K}_{ij} := n^{-2}k(c_{\vartheta(i)}, c_{\vartheta(j)})$. Since

$$\widetilde{\alpha} - \alpha = -\left[(n^{-2}\widetilde{K} + \omega^2 I)^{-1}(n^{-2}\widetilde{K} - n^{-2}K)(n^{-2}K + \omega^2 I)^{-1}\right] y,$$

we can bound

$$\begin{aligned}
\|\widetilde{\alpha} - \alpha\| &\leq \frac{\|\widetilde{K} - K\|_F \|y\|}{n\lambda_{\min}\left(\frac{1}{n^2}\widetilde{K} + \omega^2 I\right)\lambda_{\min}\left(\frac{1}{n^2}K + \omega^2 I\right)} \\
&\leq \frac{\|\widetilde{K} - K\|_F \|y\|}{n^2\omega^4}.
\end{aligned}$$

Write m as $m(x) = \frac{1}{n}\sum_{i=1}^n \alpha_i k(x_i, x) = \alpha^T k_x$, where k_x is a vector representation of the kernel evaluation, and $\hat{m}^\sigma$ as $\hat{m}^\sigma(x) = \frac{1}{n}\sum_{i=1}^n \widetilde{\alpha}_i k(c_{\vartheta(i)}, x) = \alpha^T \widetilde{k}_x$. We can now write

$$\begin{aligned}
|m(x) - \hat{m}^\sigma(x)| &= \left\|\alpha^T k_x - \widetilde{\alpha}^T \widetilde{k}_x\right\| \\
&= \left\|\alpha^T k_x - \widetilde{\alpha}^T k_x + \widetilde{\alpha}^T k_x - \widetilde{\alpha}^T \widetilde{k}_x\right\| \\
&\leq \|\alpha - \widetilde{\alpha}\| \, \|k_x\| + \|\widetilde{\alpha}\| \|k_x - \widetilde{k}_x\|.
\end{aligned}$$

Our goal is to bound these terms separately. Since κ is the maximum of the kernel, $\|k_x\| \leq \frac{\kappa}{\sqrt{n}}$, and $\|y\| \leq \sqrt{n}M$, the first term can be bounded as

$$\begin{aligned}
\|\alpha - \widetilde{\alpha}\| \, \|k_x\| &\leq \frac{\kappa}{\sqrt{n}} \|\alpha - \widetilde{\alpha}\| \\
&\leq \frac{\kappa \|y\| \|\widetilde{K} - K\|_F}{n\sqrt{n}\omega^4} \\
&\leq \frac{\kappa M \|\widetilde{K} - K\|_F}{n\omega^4}.
\end{aligned}$$

The Gaussian kernel has the property

$$(k(a,b) - k(c,d))^2 \leq \frac{1}{\sigma^2}(\|a-b\|^2 + \|b-d\|^2), \tag{6.4.12}$$

where σ is the bandwidth of the kernel. It can be shown that

$$\begin{aligned}
\|\widetilde{K} - K\|_F^2 &= \sum_{i=1}^{\tau}\sum_{j=1}^{\tau}(k(\bar{x}_i, \bar{x}_j) - k(c_{\vartheta i}, c_{\vartheta j})) \\
&= \sum_{i=1}^{\tau}\sum_{j=1}^{\tau}\left(2\left\langle \psi(\bar{x}_i) - \psi(c_{\vartheta i}), \psi(c_{\vartheta i})\right\rangle_{\mathcal{H}}\right)^2 \\
&\leq 4\tau^2\kappa^2 k_{\max},
\end{aligned}$$

where $k_{\max} := \max_i \left\|\psi(\bar{x}_i) - \psi(c_{\vartheta(i)})\right\|_{\mathcal{H}}$ is the greatest kernel approximation error. To prove a bound on the other quantity, note that

$$\begin{aligned}
\|\widetilde{\alpha}\| &\leq \left\|\left(\frac{1}{n^2}\widetilde{K} + \omega I\right)^{-1}\right\| \|y\| \\
&\leq \frac{\sqrt{n}M}{\omega^2}.
\end{aligned}$$

Furthermore, note that

$$\begin{aligned}
\left\|k_x - \widetilde{k}_x\right\| &= \frac{1}{n}\left\|\begin{pmatrix} k(x_1,x) - k(c_{\vartheta(1)},x) \\ \vdots \\ k(x_n,x) - k(c_{\vartheta(n)},x)\end{pmatrix}\right\| \\
&= \frac{\sqrt{n}}{n}\max_i \left|\left\langle \psi(x_i) - \psi(c_{\vartheta(i)}), \psi(x)\right\rangle_{\mathcal{H}}\right| \\
&\leq \frac{\kappa}{\sqrt{n}}\max_i \left\|\psi(x_i) - \psi(c_{\vartheta(i)})\right\|_{\mathcal{H}} \\
&\leq \frac{\kappa k_{\max}}{\sqrt{n}},
\end{aligned}$$

Putting the two terms together gives

$$\|\widetilde{\alpha}\|\|k_x - \widetilde{k}_x\| \leq \frac{\kappa k_{\max} M}{\omega^2},$$

which proves the theorem. □

This bound is quite weak, because it depends on all of the data the control system has seen in the past. We now prove a stronger result, that bounds the mean approximation error in a local region of the system's current state.

Theorem 6.4.4. *(**Local Approximation Theorem**) Let h and $\widetilde{h}$ be defined as above, let $\sup_{x' \in \Omega} k(x', x) \leq \kappa$, let the current state be given by x, and let $B_r(x)$ denote the ball of radius r centered at x. Denote the maximum observation in y in $B_r(x)$ by M', and the global maximum by M. Divide the dataset $\mathcal{X}$ into subsets $\mathcal{X}_1 = \{x_1, \ldots, x_{n_1}\}$, $x_i \in B_r(x)$ and $\mathcal{X}_2 = \{\bar{x}_1, \ldots, \bar{x}_{n_2}\}$, $\bar{x} \notin B_r(x)$. Generate center partitions $\mathrm{C}_1 = \{c_{\vartheta(1)}, \ldots, c_{\vartheta(n_1)}\}$ and $\mathrm{C}_2 = \{\bar{c}_{\vartheta(1)}, \ldots, \bar{c}_{\vartheta(n_2)}\}$ in a similar manner from the quantized center set C. Then*

$$\|\epsilon_m^\sigma(x)\| \leq \mathcal{O}\left(\frac{n_1}{n} M' k_{\max}\right) + \mathcal{O}\left(\frac{1}{\sqrt{n}} M\zeta\right) + \mathcal{O}\left(M\zeta\right) + \mathcal{O}\left(\frac{n_1}{n} M' \sqrt{k_{\max}}\right), \quad (6.4.13)$$

where $k_{\max} := \max_i \left\|\psi(x_i) - \psi(c_{\vartheta(i)})\right\|_{\mathcal{H}}$ is the greatest local kernel approximation error, and

$$\zeta := \max\left\{\max_{\bar{x}' \in \mathcal{X}_2} \left\langle \psi(x), \psi(\bar{x}')\right\rangle_{\mathcal{H}}, \max_{\bar{c}' \in S_2} \left\langle \psi(x), \psi(\bar{c}')\right\rangle_{\mathcal{H}}\right\}.$$

Proof. Suppose the current set of data is denoted by $\mathcal{X} = \{x_1, \ldots, x_n\}$. Divide this set into components $\mathcal{X}_1 = \{x_1, \ldots, x_{n_1}\}$, $x_i \in B_r(x)$ and $\mathcal{X}_2 = \{\bar{x}_1, \ldots, \bar{x}_{n_2}\}$, $\bar{x} \notin B_r(x)$. Generate center partitions $\mathrm{C}_1 = \{c_{\vartheta(1)}, \ldots, c_{\vartheta(n_1)}\}$ and $\mathrm{C}_2 = \{\bar{c}_{\vartheta(1)}, \ldots, \bar{c}_{\vartheta(n_2)}\}$ in a similar manner from C. First, we have that

$$\begin{aligned}
|m(x) - \hat{m}^\sigma(x)| &= \frac{1}{n}\left|\sum_{i=1}^{n} \alpha_i k(x_i, x) - \sum_{i=1}^{n} \widetilde{\alpha}_i k(c_{\vartheta(i)}, x)\right| \\
&\leq \frac{1}{n}\left|\sum_{i=1}^{n_1} \alpha_i k(x_i, x) - \sum_{i=1}^{n_1} \widetilde{\alpha}_i k(c_{\vartheta(i)}, x)\right| \\
&\quad + \frac{1}{n}\left|\sum_{j=n_1+1}^{n} \alpha_j k(\bar{x}_j, x) - \sum_{j=n_1+1}^{n} \widetilde{\alpha}_j k(\bar{c}_{\vartheta(j)}, x)\right|.
\end{aligned}$$

If $x \in B_r(x)$, then

$$\begin{aligned}\frac{1}{n}\left|\sum_{j=n_1+1}^{n}\alpha_j k(\bar{x}_j,x)-\sum_{j=n_1+1}^{n}\widetilde{\alpha}_j k(\bar{c}_{\vartheta(j)},x)\right| &\leq \frac{1}{n}\left|\sum_{j=n_1+1}^{n}\alpha_j k(\bar{x}_j,x)-\sum_{j=n_1+1}^{n}\widetilde{\alpha}_j k(\bar{x}_j,x),x\right| \\ &\quad+\frac{1}{n}\left|\sum_{j=n_1+1}^{n}\widetilde{\alpha}_j k(\bar{x}_j,x)-\sum_{j=n_1+1}^{n}\widetilde{\alpha}_j k(\bar{c}_{\vartheta(j)},x)\right| \\ &\leq \frac{1}{n}\left|\sum_{j=n_1+1}^{n}(\alpha_j-\widetilde{\alpha}_j)\zeta_1\right|+\frac{1}{n}\left|\sum_{j=n_1+1}^{n}\widetilde{\alpha}_j\zeta_2\right| \\ &\leq \frac{1}{n}\left(\zeta_1\left\|\alpha-\widetilde{\alpha}\right\|+\zeta_2\left\|\widetilde{\alpha}\right\|\right),\end{aligned}$$

where

$$\zeta_1 := \max_{\bar{x}'\in\mathcal{X}_2}\left\langle\psi(x),\psi(\bar{x}')\right\rangle_{\mathcal{H}} \tag{6.4.14}$$

$$\zeta_2 := 2\max\left\{\max_{\bar{x}'\in\mathcal{X}_2}\left\langle\psi(x),\psi(\bar{x}')\right\rangle_{\mathcal{H}},\max_{\bar{c}'\in S_2}\left\langle\psi(x),\psi(\bar{c}')\right\rangle_{\mathcal{H}}\right\}. \tag{6.4.15}$$

From the proof of the global approximation theorem,

$$\begin{aligned}\frac{1}{n}\left|\sum_{j=n_1+1}^{n}\alpha_j k(\bar{x}_j,x)-\sum_{j=n_1+1}^{n}\widetilde{\alpha}_j k(\bar{c}_{\vartheta(j)},x)\right| &\leq \frac{1}{n}\zeta_1\left(\frac{2\kappa^2 M\sqrt{n}\sqrt{k_{\max}}}{\omega^4}\right)+\frac{1}{n}\zeta_2\left(\frac{\sqrt{n}M}{\omega^2}\right) \\ &\leq \frac{1}{\sqrt{n}}\frac{2\kappa^2 M\sqrt{k_{\max}}}{\omega^4}\zeta_1+\frac{1}{\sqrt{n}}\frac{M}{\omega^2}\zeta_2 \\ &\leq \mathcal{O}\left(\frac{M\zeta}{\sqrt{n}}\right).\end{aligned}$$

This computation shows that the error associated to the kernel functions centered at the data outside $B_r(x)$ is negligible. We would now like to bound the error between the means based on the data lying within the ball, in a manner that mitigates the effect of the global maximum value M. To do this, we first note that the partitioning of the data allows us to write the kernel matrices in block matrices of the form

$$K=\begin{bmatrix}K^{11} & K^{12}\\ K^{21} & K^{22}\end{bmatrix} \quad \widetilde{K}=\begin{bmatrix}\widetilde{K}^{11} & \widetilde{K}^{12}\\ \widetilde{K}^{21} & \widetilde{K}^{22}\end{bmatrix},$$

where $K^{11}_{ij} := k(x_i,x_j)$, $K^{12}_{ij} := k(x_i,\bar{x}_j)$, and $K^{22}_{ij} := k(\bar{x}_i,\bar{x}_j)$. The matrix partitions

for $\widetilde{K}$ are defined similarly. We consider approximate linear systems of the form

$$\begin{aligned}
\alpha_B &= \begin{bmatrix} n^{-2}K^{11} + \omega^2 I_{n_1} & 0 \\ 0 & n^{-2}K^{22} + \omega^2 I_{n_2} \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\
&= \begin{bmatrix} \left(n^{-2}K^{11} + \omega^2 I_{n_1}\right)^{-1} y_1 \\ \left(n^{-2}K^{22} + \omega^2 I_{n_2}\right)^{-1} y_2 \end{bmatrix} \\
&= \begin{bmatrix} \alpha_B^1 \\ \alpha_B^2 \end{bmatrix}.
\end{aligned}$$

The vector $\widetilde{\alpha}_B$ can be computed in a similar fashion. Using the triangle inequality, we have

$$\begin{aligned}
\frac{1}{n}\left|\sum_{i=1}^{n_1} \alpha_i k(x_i, x) - \sum_{i=1}^{n_1} \widetilde{\alpha}_i k(c_{\vartheta(i)}, x)\right| &\leq \frac{1}{n}\left|\sum_{i=1}^{n_1}\left(\alpha_i k(x_i, x) - \alpha_{B_i^1} k(x_i, x)\right)\right| \\
&\quad + \frac{1}{n}\left|\sum_{i=1}^{n_1}\left(\alpha_{B_i^1} k(x_i, x) - \widetilde{\alpha}_i k(c_{\vartheta(i)}, x)\right)\right| \\
&\leq \frac{1}{n}\left\|\alpha - \alpha_{B^1}\right\| \|k_x\| \\
&\quad + \frac{1}{n}\left|\sum_{i=1}^{n_1}\left(\alpha_{B_i^1} k(x_i, x) - \widetilde{\alpha}_i k(c_{\vartheta(i)}, x)\right)\right|,
\end{aligned}$$

where k_x is the vectorized version of the kernel evaluations $k(x_i, x)$. We have

$$\begin{aligned}
\alpha - \alpha_B = &\begin{bmatrix} n^{-2}K^{11} + \omega^2 I_{n_1} & n^{-2}K^{12} \\ n^{-2}K^{21} & n^{-2}K^{22} + \omega^2 I_{n_2} \end{bmatrix}^{-1} y \\
&- \begin{bmatrix} n^{-2}K^{11} + \omega^2 I_{n_1} & 0 \\ 0 & n^{-2}K^{22} + \omega^2 I_{n_2} \end{bmatrix}^{-1} y.
\end{aligned}$$

Define E as the difference between the two matrices. We can use the proof of the

global approximation theorem to compute

$$\begin{aligned}\|\alpha - \alpha_B\| &\leq \frac{\|E\|_F \|y\|}{n\omega^4} \\ &\leq \frac{M\|E\|_F}{\sqrt{n}\omega^4} \\ &\leq \sqrt{\frac{2n_1 n_2}{n}} \frac{M\zeta_1}{\omega^4} \\ &\leq \mathcal{O}\left(\sqrt{n} M\zeta_1\right),\end{aligned}$$

which leads to the bound

$$\frac{1}{n}\|\alpha - \alpha_B\|\|k_x\| \leq \mathcal{O}\left(M\zeta_1\right),$$

which is negligible. To bound the remainder, we have that

$$\begin{aligned}\left|\sum_{i=1}^{n_1} \left(\alpha_{B_i} k(x_i, x) - \widetilde{\alpha}_i k(c_{\vartheta(i)}, x)\right)\right| \leq \|\alpha_B\| \left\|k_x - \widetilde{k}_x\right\| + \|\alpha_B - \widetilde{\alpha}_B\| \left\|\widetilde{k}_x\right\| \\ + \|\widetilde{\alpha}_B - \widetilde{\alpha}\| \left\|\widetilde{k}_x\right\|.\end{aligned}$$

We can bound each of these terms individually. Since we only consider those terms in α_B corresponding to the data within the ball, we have

$$\begin{aligned}\|\alpha_B\| \left\|k_x - \widetilde{k}_x\right\| &\leq \left(\frac{\sqrt{n_1} M'}{\omega^2}\right) \sqrt{n_1}\kappa k_{\max}, \\ &\leq \frac{n_1 M' \kappa k_{\max}}{\omega^2} \\ &\leq \mathcal{O}\left(n_1 M' k_{\max}\right).\end{aligned}$$

where M' is the largest observed value within $B_r(x)$. From $\|\widetilde{k}_x\| \leq \kappa\sqrt{n_1}$, we have

$$\|\widetilde{\alpha}_B - \widetilde{\alpha}\| \left\|\widetilde{k}_x\right\| \leq \mathcal{O}\left(nM\zeta_1\right).$$

We can apply the global approximation error theorem to the coefficients of the block matrix to get

$$\|\alpha_B - \widetilde{\alpha}_B\| \left\|\widetilde{k}_x\right\| \leq \mathcal{O}\left(\frac{n_1}{n} M' \sqrt{k_{\max}}\right).$$

Putting these together yields

$$\frac{1}{n}\left|\sum_{i=1}^{n_1}\left(\alpha_{B_i}k(x_i,x)-\widetilde{\alpha}_i k(c_{\vartheta(i)},x)\right)\right| \leq \mathcal{O}\left(\frac{n_1}{n}M'k_{\max}\right)+\mathcal{O}\left(M\zeta_1\right)+\mathcal{O}\left(\frac{n_1}{n}M'\sqrt{k_{\max}}\right),$$

which proves the theorem. □

Note the trade off implicit in the bound (6.4.13): the greater the radius of the ball $B_r(x)$, the smaller ζ is, which implies that points seen outside $B_r(x)$ have minimal effect on the estimation of the GP mean. However, the larger the radius, the greater the number of values within $B_r(x)$, and therefore M' is chosen to be the maximum value of a much larger set.

The boundedness of the tracking error can now be proven.

Theorem 6.4.5. *Consider the system in (6.2.1), the control law of (6.2.2) and (6.2.6), and assume that the uncertainty $\Delta(\bar{x})$ is representable by a Gaussian process (6.3.1). Assume that $\|\epsilon_m^\sigma(\bar{x}_0)\|$ is bounded for the initial state $\bar{x}_0$. Then Algorithm 13 and the adaptive signal $\nu_{ad}(\bar{x})=\hat{m}^\sigma(z)$ guarantees that the system is mean square uniformly ultimately bounded a.s.*

Proof. Let $V(e(t))=\frac{1}{2}e^T(t)Pe(t)$ be the stochastic Lyapunov candidate, where $P>0$ satisfies the Lyapunov equation (6.2.9). Note that the Lyapunov candidate is bounded above and below by a quadratic term since $\frac{1}{2}\lambda_{\min}(P)\|e\|^2 \leq V(e) \leq \frac{1}{2}\lambda_{\max}(P)\|e\|^2$. The Itô differential of the Lyapunov candidate along the solution of (6.4.8) for the σ^{th} system is

$$\begin{aligned}\mathcal{L}V(e) &= \sum_i \frac{\partial V(e)}{\partial e_i}Ae_i+\frac{1}{2}\sum_{i,j}\left[BG_\sigma(BG_\sigma)^T\right]_{ij}\frac{\partial^2 V(e)}{\partial e_j \partial e_i}\\ &= -\frac{1}{2}e^TQe+e^TPB[\epsilon_m^\sigma(z)+G_\sigma\, d\xi(t)]+\frac{1}{2}\mathrm{tr}\left(BG_\sigma(BG_\sigma)^TP\right).\end{aligned}$$

Let $c_1 := \frac{1}{2}\|P\|\|BG_\sigma\|^2$ and $c_2 := \|PB\|$, then

$$\begin{aligned}\mathcal{L}V(e) &\leq -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 + c_2\|e\|\|\epsilon_m^\sigma(\bar{x}) + G_\sigma\, d\xi(t)\| + c_1 \\ &\leq -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 + c_2\|e\|(\|\epsilon_m^\sigma(\bar{x})\| + \|G_\sigma\, d\xi(t)\|) + c_1 \\ &\leq -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 + c_2\|e\|\,(\|\epsilon_m^\sigma(\bar{x})\| + c') + c_1 \end{aligned} \tag{6.4.16}$$

From Theorem 6.4.3, $\|\epsilon_m^\sigma(\bar{x})\| \leq c_3 M^\sigma \sqrt{k_{\max}^\sigma} + c_4 M^\sigma k_{\max}^\sigma$, so

$$\mathcal{L}V(e) \leq -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 + c_2\|e\|\,(c_3\Upsilon^\sigma + c_4 M^\sigma k_{\max}^\sigma + c') + c_1.$$

Define

$$c_5 := c_3\sqrt{k_{\max}^\sigma} + c_4 k_{\max}^\sigma. \tag{6.4.17}$$

Therefore outside of the set,

$$\Theta_\gamma = \left\{ \|e\| \geq \frac{c_5 M^\sigma + \sqrt{(c_5 M^\sigma)^2 + 2\lambda_{\min}(Q)c_1}}{\lambda_{\min}(Q)} \right\}, \tag{6.4.18}$$

$\mathcal{L}V(e) \leq 0$ a.s. It is assumed that M^σ is initially bounded. The RHS of (6.4.18) determines the size of Θ_γ. For a fixed budget $p_{\max}$, the estimated GP mean $\hat{m}^\sigma(z)$ is an RBF network. Since we employ the Gaussian kernel, Therefore, we can assume that the size of Θ_γ is fixed.

Let $\beta = \max_{e\in\theta_\gamma} V(e)$ and define the compact set $\Theta_\beta = \{e : \mathcal{L}V(e) \leq \beta\}$. Note that $\Theta_\gamma \subseteq \Theta_\beta$ with $\mathcal{L}V(e) < 0$ a.s. outside of Θ_β. Furthermore, note that $\frac{1}{2}\lambda_{\min}(P)\|e\|^2 \leq V(e)$ and define $\delta = \left(\frac{2\beta}{\lambda_{\min}(P)}\right)^{\frac{1}{2}}$. Then, with $r(t)$ such that x_{rm} remains bounded within $\mathcal{B}_m$, $x(t) \in \mathcal{D}$ a.s. and the solution to (6.2.1) holds. Since this is true for all σ, and because Algorithm 13 guarantees that σ does not switch arbitrarily fast, by Theorem 6.4.1, (6.4.8) is mean square uniformly ultimately bounded inside of this set a.s. [44]. □

Remark 2. *In contrast to traditional parametric adaptive control, the Lyapunov candidate does not need to be an explicit function of the adaptive element's parameters. This is because the parameters of the budgeted nonparametric mean function are*

discretely updated and guaranteed to be bounded over every switching interval because they either stay constant or are "renormalized". Furthermore, note that the size of the set within which e is bounded can be reduced by reducing the representation error $\epsilon_m^\sigma(\bar{x})$ by choosing smaller values of ϵ_{tol}, or increasing $\lambda_{\min}(Q)$ by appropriately selecting K_1, K_2 in (6.2.6). Finally, in [47], it was shown that the linear independence of $\mathcal{BV}$ ensures that persistency of excitation (PE) in the state space is visible in $\mathcal{H}$. Since the KL variant of the algorithm aims to enforce this independence subject to the tolerance ϵ_{tol}, PE is never lost (ensuring $K(Z_\tau, Z_\tau)$ is invertible).

Remark 3. *The assertions in Theorem 6.4.5 are stronger than those typically proven for fixed-center RBFN-MRAC [46, 85] because a domain over which the centers have been allocated is not required to be assumed, rather the operating domain $\mathcal{D}$ over which conditions to guarantee a continuous time Markov process solution to (6.4.1) are satisfied is simply required to exist.*

The following corollary extends the above result to the case when an exact representation of the mean function is available (i.e. $\epsilon_m(z) = 0$), such as might be possible in cases where the uncertainty can be represented accurately using a tabular representation, and is equivalent to assuming that the allowable function class in (6.3.2) contains a finite number of kernels. This corollary is also applicable to robust control of linear systems where the uncertainty is characterized by a GP whose mean has been identified using recorded data.

Corollary 6.4.2. *Consider the system in (6.2.1), the reference model in (6.2.5), and the control law of (6.2.2) and (6.2.6). Let the uncertainty $\Delta(z)$ follows a Gaussian process as in (6.3.1) with a fixed number of kernels, then Algorithm 13 and the adaptive signal $\nu_{ad}(z) = m(z)$ guarantees that $e(t)$ is exponentially mean squared ultimately bounded.*

Proof. With $\epsilon_m(z) = 0$, (6.4.16) reduces to

$$\mathcal{L}V(e) \leq -\frac{\lambda_{\min}(Q)}{2\lambda_{\max}(P)}V(e) + c_4. \tag{6.4.19}$$

The result follows from Theorem 6.4.1 with constant σ (no switching). □

The following corollary shows that if $\epsilon_m^\sigma(\bar{x}) = \hat{m}^\sigma(\bar{x}) - m(\bar{x})$ tends to zero as $t \to \infty$, then the closed loop system is uniformly exponentially ultimately bounded a.s. This result is useful in the case where the mean $m(\bar{x})$ is exactly a finite linear combination of l kernels and an online GP learning algorithm eventually learns an exact representation of the mean function given sufficient samples.

Corollary 6.4.3. *Consider the system in (6.2.1), the reference model in (6.2.5), the control law of (6.2.2) and (6.2.6). Let the uncertainty $\Delta(z)$ follow a Gaussian process as in (6.3.1). Then Algorithm 13 and the adaptive signal $\nu_{ad}(z) = \hat{m}^\sigma(\bar{x})$ with $\lim_{t\to\infty} \hat{m}^\sigma(\bar{x}) = m(\bar{x})$ guarantees that $e(t)$ is mean squared ultimately bounded.*

Proof. Since $\epsilon_m(z) \to 0$ as $t \to \infty$ we have that (6.4.16) approaches (6.4.19). The result now follows from Theorem 6.4.1. □

6.5 Trajectory Tracking in Presence of Wing Rock Dynamics in an unknown operating domain

In this section the GP-MRAC approach is compared with fixed-center RBFN-MRAC in numerical simulations on the wing rock phenomenon. Note that the goal is not to present a highly tuned adaptive controller for wing rock, several other authors have already done that assuming known operating domain or known bases of $\Delta(x)$ [20, 43, 61, 93, 114]. Rather, the goal is to test the performance when the assumption of known operating domain (needed by fixed-center RBFN-MRAC to pre-allocate centers) or known bases of uncertainty are violated, forcing the controller to adapt to unknown operating conditions. In that sense, one would be tempted to compare the results to Chapter 5, but the results here are not directly comparable because that method does not consider stochasticity in the dynamics and measurement noise, as is done here.

Let θ denote the roll attitude of an aircraft, p denote the roll rate and δ_a denote the aileron control input. As before, a model for wing rock dynamics is [61]

$$\begin{aligned} \dot{\theta} &= p \\ \dot{p} &= L_{\delta_a}\delta_a + \Delta(x), \end{aligned} \tag{6.5.1}$$

where $L_{\delta_a} = 3$ and $\Delta(x)$ is a model of the wing rock dynamics and is assumed to be unknown to the controller. Unlike [84] and [61] who assume a deterministic model with a known basis for $\Delta(x)$, we assume that the uncertainty arises from a distribution over functions approximated by a GP with variance ω_n^2 and mean

$$\bar{\Delta}(x) = W_0^* + W_1^*\theta + W_2^* p + W_3^*|\theta|p + W_4^*|p|p + W_5^*\theta^3. \tag{6.5.2}$$

The parameters for the mean function and the settings for the control law and reference model are the same as those in §5.6. Stochasticity is introduced by adding Gaussian white noise of variance $\omega_n = 0.01$ to the states. The maximum number of points to be stored ($p_{\max}$) was arbitrarily set to 100, and points were selected for storage based on both the oldest point (OP) and KL divergence (KL) (see Section 6.3.2). The projection operator places an upper bound of 10 on individual RBF weights when using RBFN-MRAC with fixed centers, see (5.3.5). Both the case when the chosen RBFN centers cover the expected domain of operation and the case when the system is driven outside of the a priori expected domain of operation are examined.

6.5.1 System Within Domain of Operation

In the first case, a set of reference commands lead the system, after a short transient, to a relatively compact set in the state space, which allows for good online approximation of the uncertainty with 100 centers. The goal of this scenario is to compare GP-MRAC with fixed-center RBFN-MRAC when the domain of operation is known. Figure 32a compares the system states and the reference model when using GP-MRAC versus RBFN-MRAC with the projection operator and uniformly distributed centers over $[-2, 2] \times [-2, 2]$. While GP-MRAC performs significantly better

in the transient, especially when tracking the command in $\dot{\theta}$, the long term behavior of all the controllers is similar. This is to be expected as the reference model drives the system to operate over the known domain of operation, hence RBFN-MRAC can be tuned to yield very good tracking [43, 114]. However, the poorer performance of RBFN-MRAC in the initial transient stresses the fact that transient performance guarantees of RBFN-MRAC can be lost if the system states leave the expected domain of operation, this is not the case for the nonparametric GP-MRAC approach as it selects centers online. Figure 33a compares the tracking error for both the controllers. The GP-MRAC system has less oscillations. Figure 34a compares the learned models of GP-MRAC and RBFN-MRAC. While the GP adaptive element output is almost indistinguishable from the uncertainty in presence of measurement noise, the RBFN does not accurately learn the uncertainty. The poor learning performance of RBFN with the learning law of 5.3.5 is to be expected[8, 46, 104], and Figure 34a clearly shows that GP-MRAC yields much better learning. 35a plots the energy of the spectrum of the signal $\nu_{ad} - \Delta$ for all the controllers considered. It can be seen that for the RBFN-MRAC controller the spectra for this signal has more energy at nonzero frequencies. This indicates that there are greater number of oscillations in RBFN-MRAC. Figure 36a shows online uncertainty tracking for the KL divergence method by plotting the posterior mean and variance at each timestep, showing good approximation error, and confidence in the estimates. Figure 37a shows the trajectory of the system controlled by the KL divergence method in state space.

6.5.2 System Driven Outside Domain of Operation

The above scenario demonstrated that the baseline trajectory tracking performance of GP-MRAC is somewhat comparable to RBFN-MRAC with update law of 5.3.5 when the system stays within operating domain. However, GP-MRAC's true usefulness becomes apparent when its nonparametric form is more fully exploited. Therefore, in this second case, the system is given reference commands that drive it outside of

the expected domain of operation, i.e. where RBF centers are not allocated, and RBFN-MRAC has not been tuned to yield good performance. Significant performance differences occur between the two methods. Figures 32b and 33b show that the tracking performance of GP-MRAC is superior to RBFN-MRAC, especially when tracking the $\dot{\theta}$ commands. Figure 34b shows that RBFN-MRAC does an even poorer job of learning the uncertainty than before, while GP-MRAC learning performance is excellent. In Figure 35b, the harmonics associated to the oscillations become much more apparent. It can be clearly seen that there is a significant peak at nonzero frequencies for RBFN-MRAC that is not present in GP-MRAC. This indicates that the GP-MRAC controller is better able to mitigate the oscillations associated with wing rock dynamics. Figure 37b shows the distribution of centers chosen by the controllers. Although the KL divergence method more widely distributes the centers over the trajectory of the system in comparison to the OP method, there is little difference in the online performance between the OP and KL divergence GP-MRAC controllers. The negligible difference indicates that the most important factor to performance is the existence of centers near the current state. However, their performance differs significantly in terms of long-term learning.

6.5.3 Illustration of Long-Term Learning Effect

One of the most important contributions of GP-MRAC, particularly when using the KL divergence scheme, is the introduction of long-term learning in the adaptive controller. Long-term learning here is characterized by better approximation of the uncertainty over the entire domain, a quality that is missing from RBFN-MRAC with the update law of 5.3.5. In order to illustrate this, the learned parameters of the adaptive elements of the systems are recorded at the final time T of the simulation. The systems are then reinitialized with these parameters, which are *frozen*, and the simulations are run again. Figures 38a and 38b show the uncertainty tracking results for the two scenarios mentioned above. As can be seen, the locality that drives the

RBFN-MRAC updates results in a controller that doesn't capture the uncertainty globally. While GP-MRAC with the OP scheme does a better job in this regard, its choice of center placement means that approximation accuracy is local also. Since the KL divergence scheme chooses points to keep based on the distance between the GP models, it is able to retain global information about the uncertainty and thus achieve the best performance.

Overall, these results show that the GP-MRAC as described by Algorithm 13 is successful in capturing the uncertainty without the need for prior domain domain knowledge.

6.6 Appendix

This section outlines the implementation details of the sparse online Gaussian process algorithm [24]. At time $\tau + 1$, given a new datapoint $z_{\tau+1}$, the algorithm minimizes the KL divergence between the model with the datapoint included, and the $\tau + 1$ models with one datapoint deleted. To compute the updates in an online fashion, define the scalar quantities

$$q^{(\tau+1)} = \frac{y - \alpha_\tau^T k_{x_\tau}}{\omega_n^2 + k_{x_\tau}^T C_\tau k_{x_\tau} + k_\tau^*}, \tag{6.6.1}$$

$$r^{(\tau+1)} = -\frac{1}{\omega_n^2 + k_{x_\tau}^T C_\tau k_{x_\tau} + k_t^*}, \tag{6.6.2}$$

where α_τ, k_{x_τ}, and C_τ are defined in (6.3.6) and (6.3.7). Let $e_{\tau+1}$ be the $(\tau + 1)$ coordinate vector, and let $T_{\tau+1}(\cdot)$ and $U_{\tau+1}(\cdot)$ denote operators that extend a τ-dimensional vector and matrix to a $(\tau + 1)$ vector and $(\tau + 1) \times (\tau + 1)$ matrix by appending zeros to them. The GP parameters can be solved for recursively using the equations

$$\begin{aligned} \alpha_{\tau+1} &= T_{\tau+1}(\alpha_\tau) + q^{(\tau+1)} s_{\tau+1}, \\ C_{\tau+1} &= U_{\tau+1}(C_\tau) + r^{(\tau+1)} s_{\tau+1} s_{\tau+1}^T, \\ s_{\tau+1} &= T_{\tau+1}(C_\tau k_{x_{\tau+1}}) + e_{\tau+1}. \end{aligned} \tag{6.6.3}$$

The inverse of the Gram matrix, dentoed by Q, needed to solve for $\gamma_{\tau+1}$ is updated online through the equation

$$Q_{\tau+1} = U_{\tau+1}(Q_\tau) + \gamma_{\tau+1}^{-1} \left(T_{\tau+1}(\hat{e}_{\tau+1}) - e_{\tau+1}\right) \left(T_{\tau+1}(\hat{e}_{\tau+1}) - e_{\tau+1}\right)^T, \tag{6.6.4}$$

where $\hat{e}_{\tau+1} := Q_\tau k_{z_{\tau+1}}$. Finally, in order to delete an element, one computes the model parameters with the $(\tau+1)$-th point, and chooses the basis vector with the smallest score measure, given by

$$\epsilon_i = \frac{|\alpha_{\tau+1}(i)|}{Q_{\tau+1}(i,i)}. \tag{6.6.5}$$

Let ι be the basis vector chosen to be discarded by the score (6.6.5). Then the deletion equations are given by

$$\begin{aligned} \hat{\alpha} &= \hat{\alpha}^{\neg\iota} - \alpha^* \frac{Q^*}{q^*}, \\ \hat{C} &= C^{\neg\iota} + c^* \frac{Q^* Q^{*T}}{{q^*}^2} - \frac{1}{q^*}\left[Q^* C^{*T} + C^* Q^{*T}\right], \\ \hat{Q} &= Q^{\neg\iota} - \frac{Q^* Q^{*T}}{q^*}, \end{aligned} \tag{6.6.6}$$

where α^* is the ι^{th} component in the vector $\alpha_{\tau+1}$, and $\alpha^{\neg\iota}$ represents the remaining vector. Similarly, $C^{\neg\iota}$ ($Q^{\neg\iota}$) represents the $\tau \times \tau$ submatrix in the $(\tau+1)\times(\tau+1)$ matrix $C_{\tau+1}$ ($Q_{\tau+1}$) associated to the basis vectors being kept, c^* (q^*) represents the (ι,ι) index into the matrix chosen by the score measure, and C^* (Q^*) is the remaining τ-dimensional column vector. Using the above equations, the budgeted sparse GP algorithm is summarized by Algorithm 14.

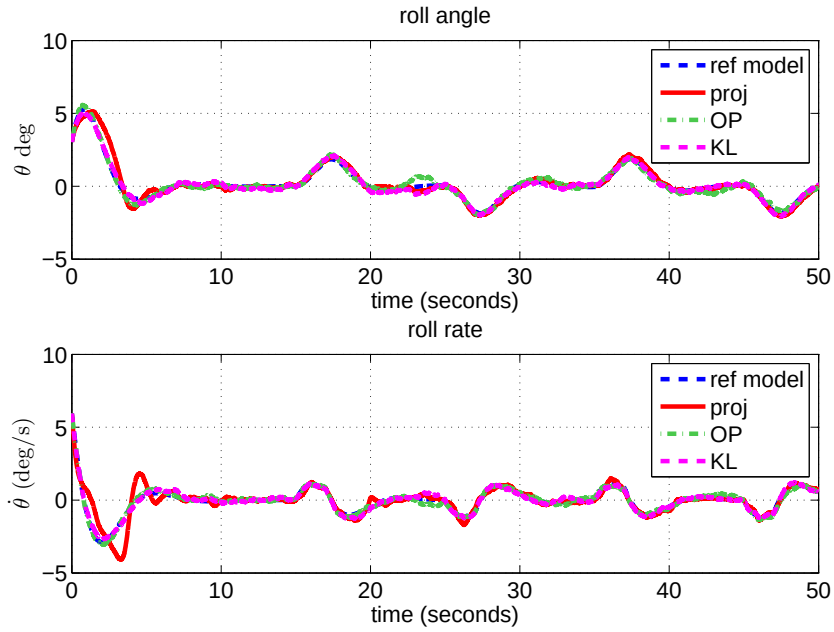
6.7 Conclusion

This paper modeled the uncertainty in Model Reference Adaptive Control (MRAC) as a distribution over functions rather than via a deterministic function. This approach to uncertainty modeling is relevant to many real-world scenarios involving the presence of noise, servo chattering, or other non-smooth effects. To accurately learn the uncertain function, we used Gaussian Process adaptive elements, which leverage

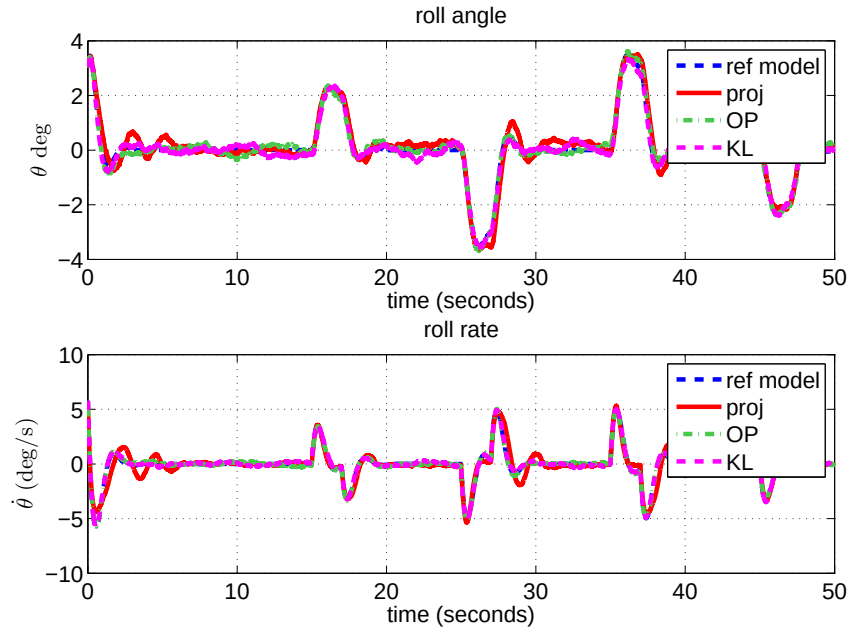
Algorithm 14 The budgeted sparse Gaussian process algorithm

while new measurements $(z_{\tau+1}, y_{\tau+1})$ are available **do**
  Compute $q^{(\tau+1)},\ r^{(\tau+1)},\ k^*_{\tau+1},\ k_{z_{\tau+1}},\ \hat{e}_{\tau+1}$ and $\gamma_{\tau+1}$.
  if $\gamma_{\tau+1} < \epsilon_{tol}$ **then**
    Perform a reduced update, using $\hat{e}_{\tau+1}$ in (6.6.3) without extending the length of the parameters α and C.
  else
    Perform the update in (6.6.3) using $e_{\tau+1}$. Add the current input to the $\mathcal{BV}$ set, and compute the Gram matrix inverse using (6.6.4).
    if $|\mathcal{BV}| > p_{\max}$ **then**
      Compute scores for the candidate $\mathcal{BV}$'s using (6.6.5), find the vector corresponding to the lowest score, and delete it using (6.6.6).
    end if
  end if
end while

a powerful and robust nonparametric framework to perform inference directly in the space of functions. We extended a GP inference method to work within a preset budget, allowing its use in an online setting. These two modifications define the GP-MRAC algorithm, which was proven to be stable through the use of stochastic stability theory for switched systems. Simulations employing GP-MRAC for the control of wing rock dynamics demonstrate the efficacy of GP-MRAC as a budgeted, nonparametric adaptive control method which requires no domain knowledge of the uncertainty.

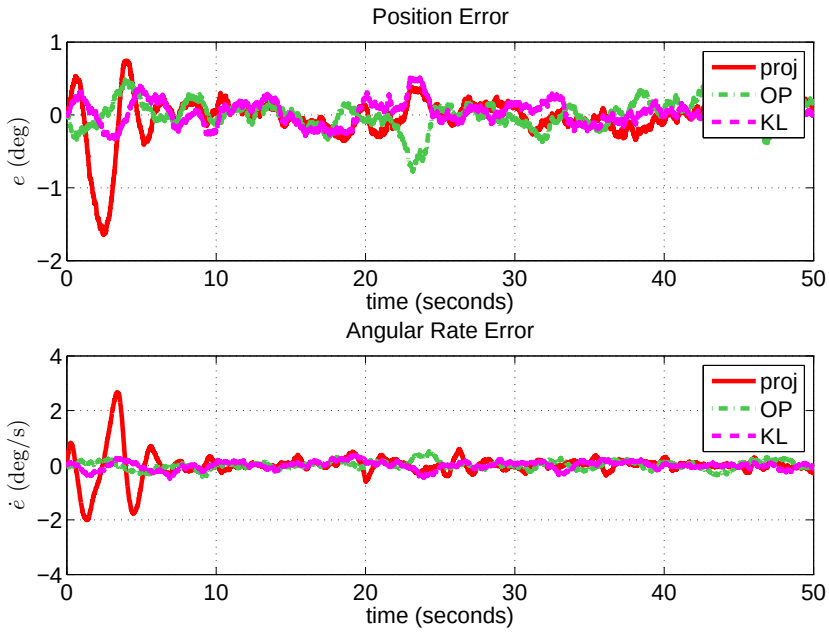


(a) Within Domain

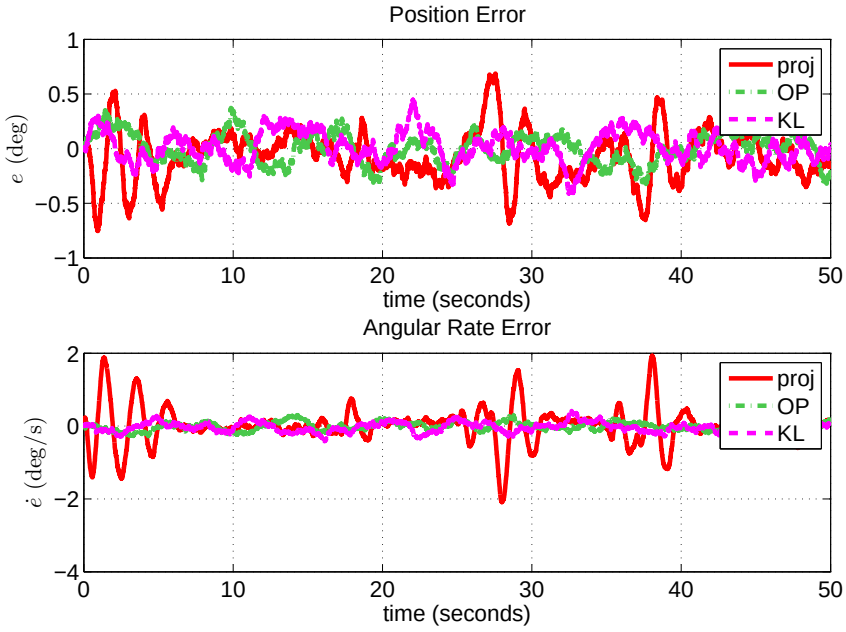


(b) Outside Domain

Figure 32: Comparison of system states and reference model when using GP regression based MRAC and RBFN-MRAC with the projection operator and uniformly distributed centers over their respective domains. The state measurements are corrupted with Gaussian white noise.

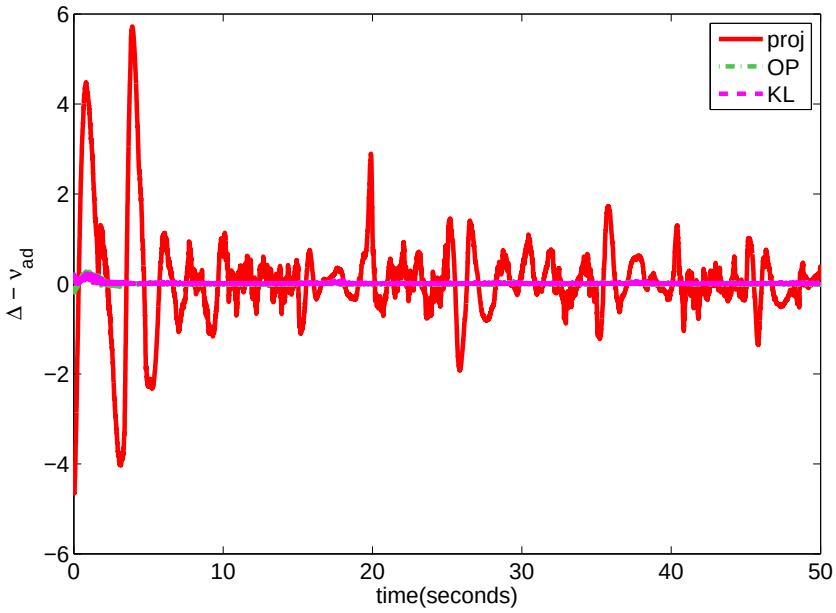


(a) Within Domain

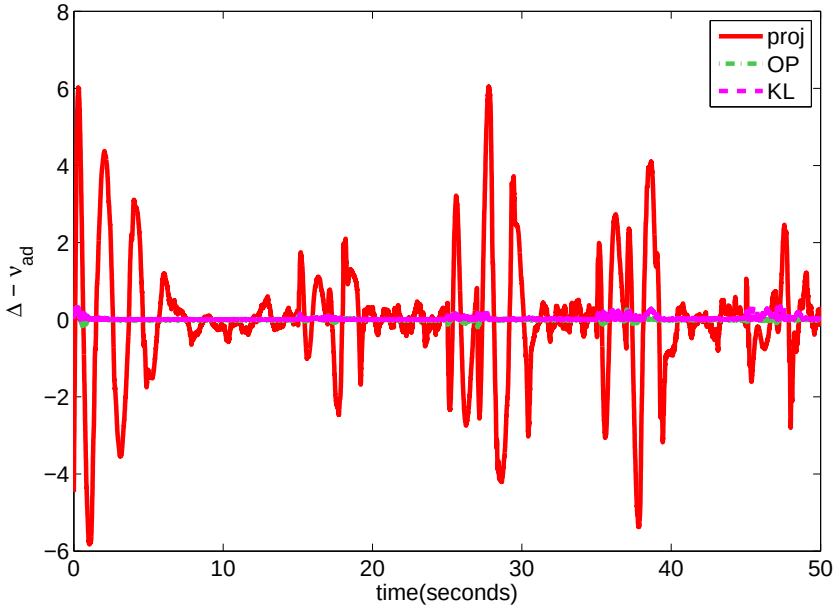


(b) Outside Domain

Figure 33: Comparison of tracking error when using GP regression based MRAC and RBFN-MRAC with the projection operator and uniformly distributed centers over their respective domains. Compared to Fig. 32a, RBFN-MRAC with uniformly distributed centers has higher transient tracking error than GP-MRAC because the commands drive the system out of the range over which the centers were distributed.

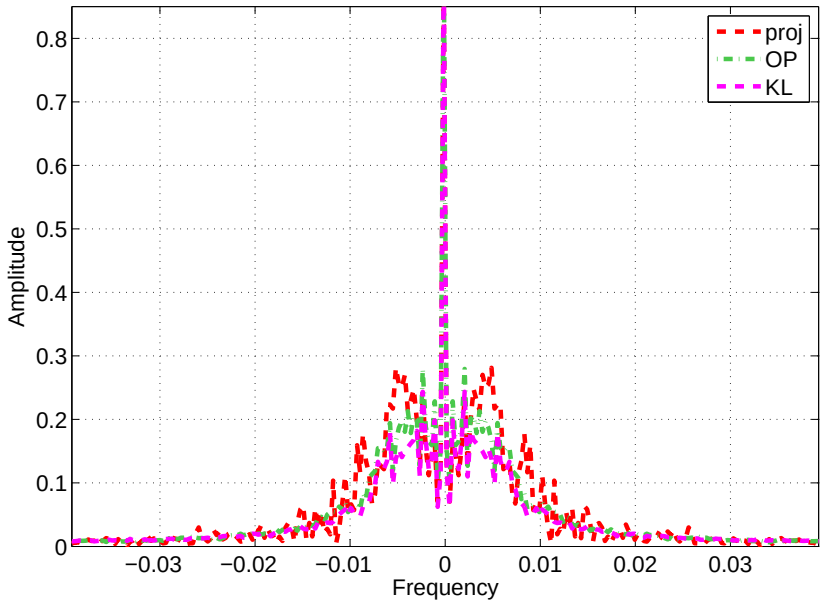


(a) Within Domain

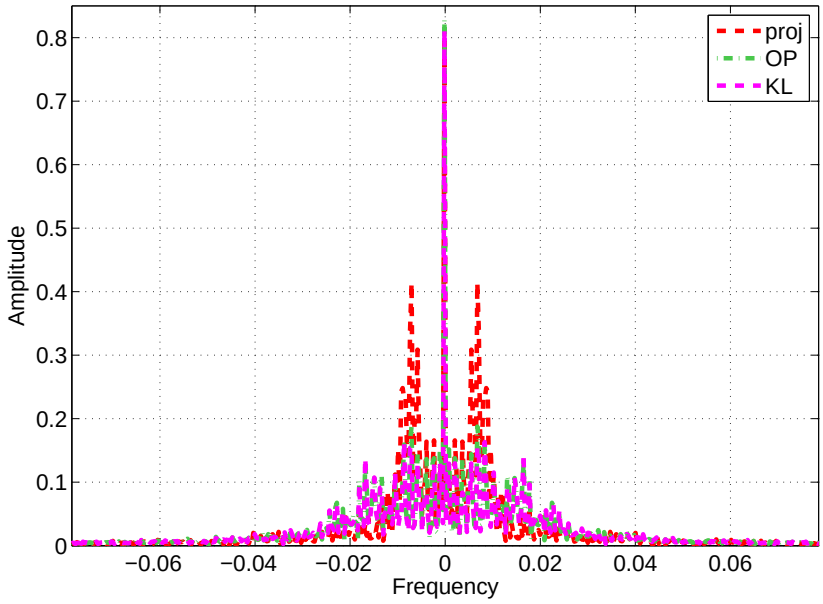


(b) Outside Domain

Figure 34: Error between the adaptive element output and the actual uncertainty (the signal $\nu_{ad} - \Delta$). RBF MRAC approximation with uniformly distributed centers is significantly worse than the GP approximations.

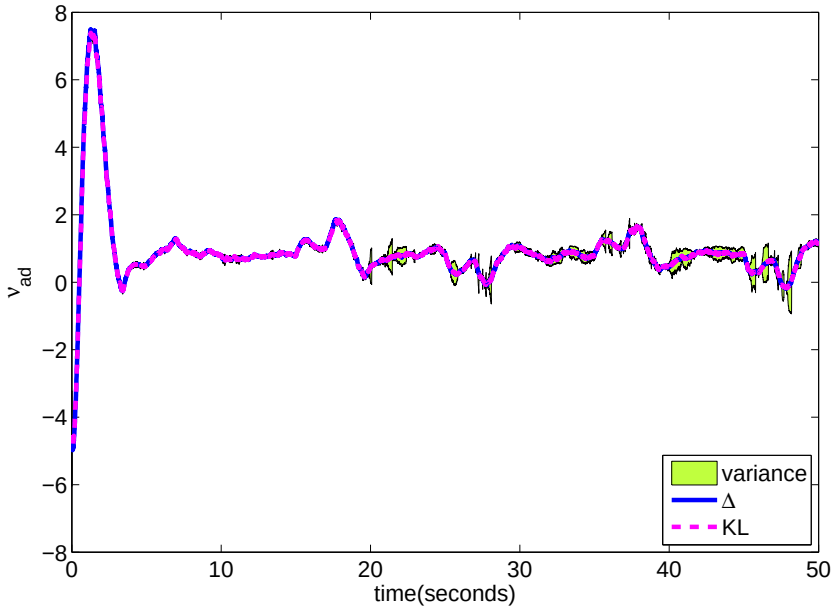


(a) Within Domain

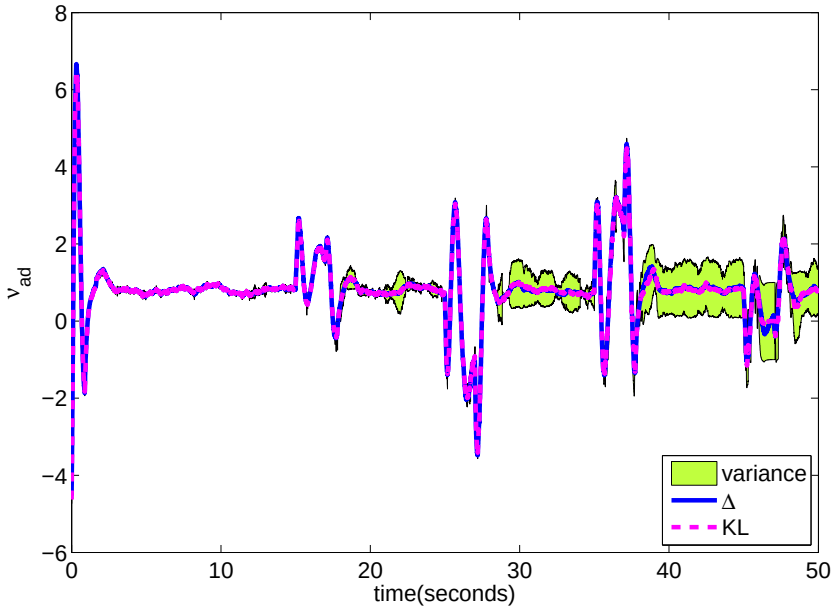


(b) Outside Domain

Figure 35: Energy of the spectra of the error between the adaptive element output and the actual uncertainty. This figure quantifies the greater number of oscillations while tracking in RBF MRAC.

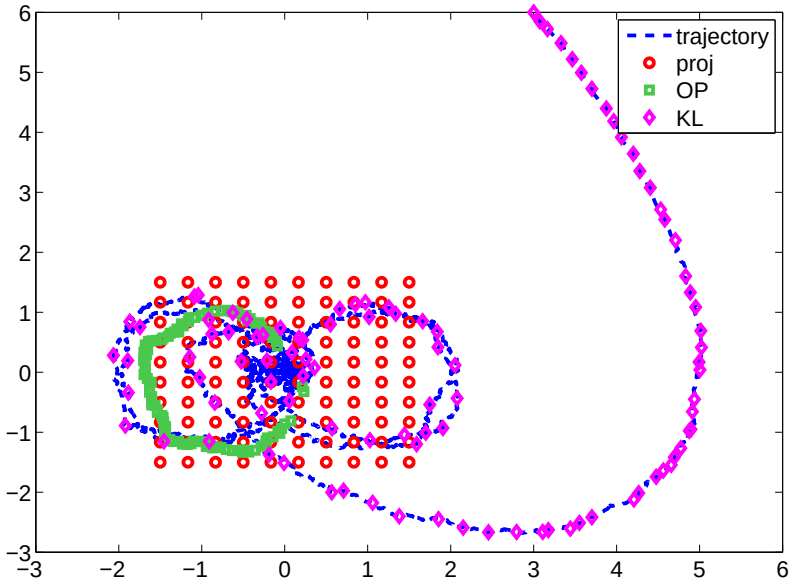


(a) Within Domain

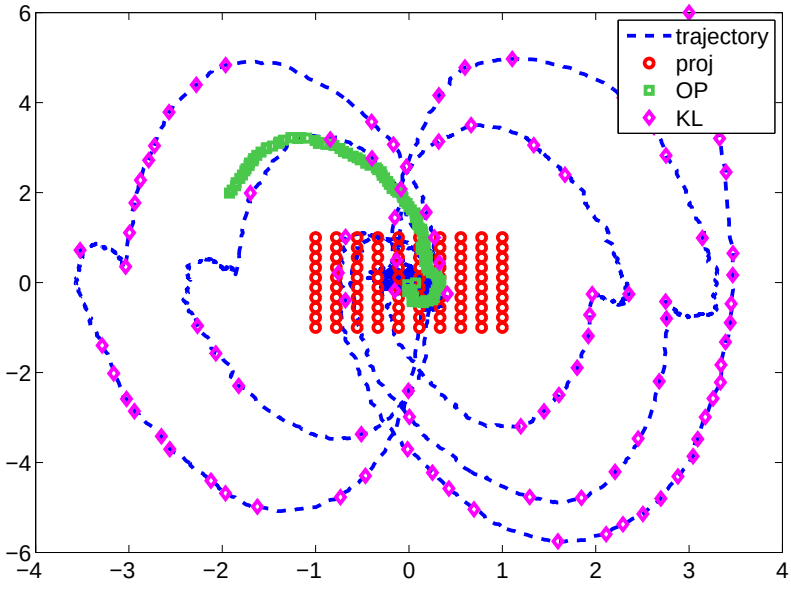


(b) Outside Domain

Figure 36: Online uncertainty tracking for the KL method.

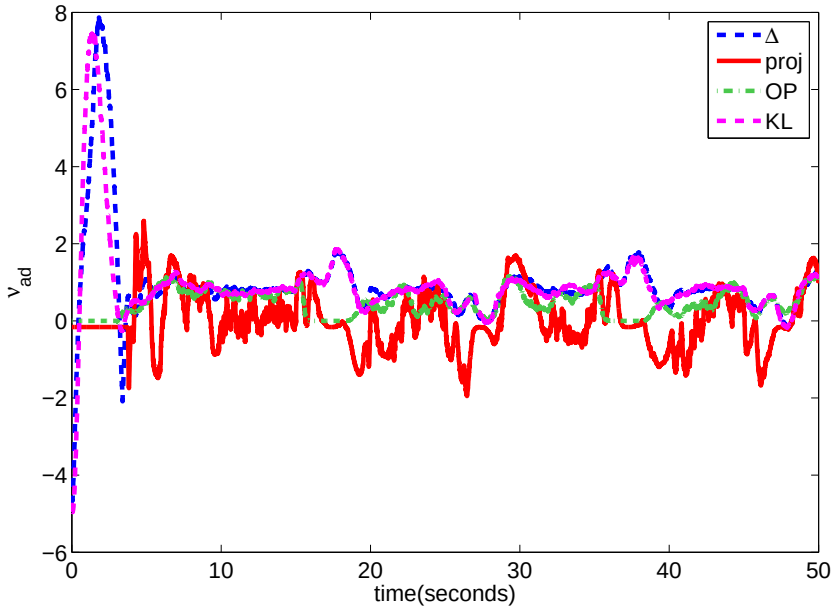


(a) Within Domain

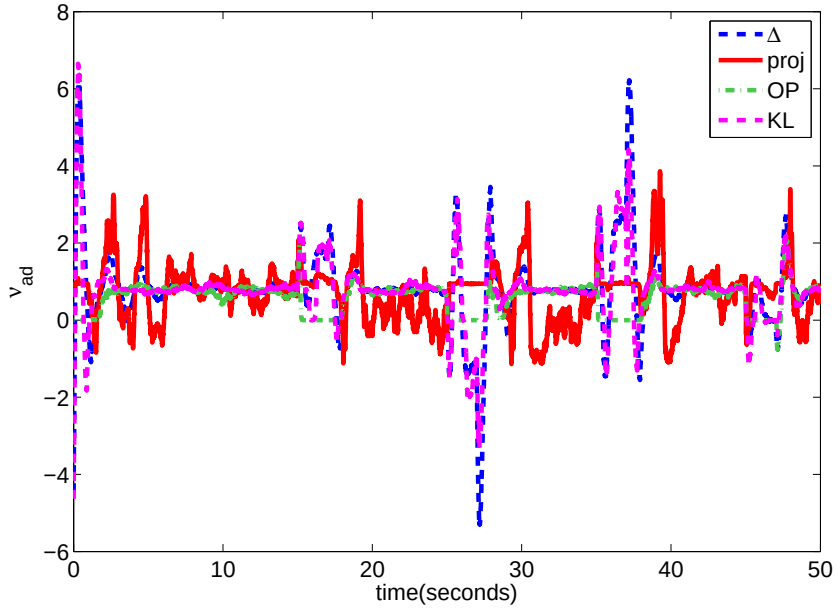


(b) Outside Domain

Figure 37: Trajectories in the state space for both command sets.



(a) Within Domain



(b) Outside Domain

Figure 38: Comparison of uncertainty tracking after the models are learned and the weights are frozen. As can be seen, the locality of the proj operator and OP controllers precludes true learning upon the domain.

CHAPTER VII

CONCLUSION

This thesis explored the role of reduced-set approximations for kernel methods in both the batch and online data cases. In the batch case, we presented approximation strategies for generating the nonlinear feature mapping associated to any kernel method in a manner that results in both training and testing speedups, and whose approximation accuracy in guaranteed to be bounded by quantities that can be computed in at most $\mathcal{O}(m)$ time, where m is the cardinality of the reduced set. In the online case, we utilized the theory behind kernel methods to design new adaptive controllers that outperform previous controllers by assuming less knowledge about the domain of the uncertainty and by instilling memory into the controller architecture. Our key contributions can be summarized as follows:

- **Reduced-set KPCA**: We formulated an approximation strategy to perform a fast KPCA. In doing so, we proved that the reduced-set density estimate generated from the approximation, under the action of the kernel, is intimately connected to the operator approximation error. For radially-symmetric kernels, we presented a particularly simple, single-pass method called the shadow density estimate, which allows the computation of bounds in closed form.

- **Reduced-set models for batch data**: We derived similar reduced-set approximations for Gaussian process regression, diffusion maps, and kernel embeddings, and proved similar theorems bounding approximation error.

- **Reduced-set models for adaptive control**: Using a connection between RKHS theory and persistency of excitation, we formulated a reduced-set kernel

method for MRAC, proved its stability, showed how it can be used to remove the need for a priori knowledge on the domain of the uncertainty, and demonstrated its effectiveness in reducing tracking error in simulations.

- **GP-MRAC**: We replaced the adaptive element in MRAC by a reduced-set approximation for Gaussian process regression, which can naturally accomodate varying modes of operation, as well as process and observation noise. We proved the stability of this controller using stochastic stability arguments, and demonstrated its effectiveness in learning and combatting both time-invariant uncertainty.

Future projects that the work in this thesis naturally leads includes the formulation of reduced-set models for kernels that are not positive-definite, such as those used in the neurons of deep learning networks, and the extension of GP-MRAC to learning nonstationary Markov decision processes.

APPENDIX A

MINIMIZATION OF THE OPERATOR ERROR FOR THE GAUSSIAN KERNEL

In this chapter, we demonstrate how to minimize the upper bound (3.5.3) for the Gaussian kernel. Suppose S_j is fixed. Then taking the derivative of the quantity (3.5.3) with respect to the center c_j results in

$$\frac{\partial}{\partial c_j}\sum_{j=1}^{m}\sum_{x_i\in S_j}\sqrt{\left(\kappa-k(x_i,c_{\vartheta(i)})\right)}=\sum_{j=1}^{m}\sum_{x_i\in S_j}\frac{\partial}{\partial c_j}\sqrt{\left(\kappa-k(x_i,c_{\vartheta(i)})\right)}.$$

For each of the terms, we have

$$\begin{aligned}\frac{\partial}{\partial c_j}\left(\kappa-k(x_i,c_{\vartheta(i)})\right)^{\frac{1}{2}}&=\frac{\partial\left(\kappa-k(x_i,c_{\vartheta(i)})\right)^{\frac{1}{2}}}{\partial\left(\kappa-k(x_i,c_{\vartheta(i)})\right)}\frac{\partial\left(\kappa-k(x_i,c_{\vartheta(i)})\right)}{\partial c_j}\\&=\frac{1}{2\sqrt{\left(\kappa-k(x_i,c_{\vartheta(i)})\right)}}\frac{\partial\left(\kappa-k(x_i,c_{\vartheta(i)})\right)}{\partial c_j}.\end{aligned}$$

Looking at the term on the RHS, we have

$$\begin{aligned}\frac{\partial\left(\kappa-k(x_i,c_{\vartheta(i)})\right)}{\partial c_j}&=-\frac{\partial k(x_i,c_{\vartheta(i)})}{\partial c_j}\\&=-\frac{e^{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}}{\partial c_j}\\&=-\frac{e^{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}}{\partial\frac{-\|x_i-c_j\|^2}{2\sigma^2}}\frac{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}{\partial c_j}\\&=-e^{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}\left(\frac{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}{\partial c_j}\right)\\&=\frac{1}{2\sigma^2}e^{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}\left(\frac{\|x_i-c_j\|^2}{\partial c_j}\right)\\&=\frac{1}{2\sigma^2}e^{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}\left(2(c_j-x_i)\right)\\&=\frac{1}{\sigma^2}e^{\frac{-\|x_i-c_j\|^2}{2\sigma^2}}(c_j-x_i).\end{aligned}$$

Plugging in all these terms together into the original equation, we have

$$\frac{\partial}{\partial c_j} \sum_{j=1}^{m} \sum_{x_i \in S_j} \sqrt{\left(\kappa - k(x_i, c_{\vartheta(i)})\right)} = 0$$

$$\sum_{j=1}^{m} \sum_{x_i \in S_j} \frac{k(x_i, c_j)}{2\sigma^2 \sqrt{(\kappa - k(x_i, c_j))}} (c_j - x_i) = 0$$

$$\sum_{j=1}^{m} \sum_{x_i \in S_j} \frac{k(x_i, c_j)}{\sqrt{(\kappa - k(x_i, c_j))}} (c_j - x_i) = 0.$$

For a k-means style update equation per centroid, define c_j^{old} as the last position of the centroid and c_j^{new} as the new position of the centroid being solved for. Then

$$\sum_{x_i \in S_j} \frac{k(x_i, c_j^{\text{old}})}{\sqrt{\left(\kappa - k(x_i, c_j^{\text{old}})\right)}} (c_j^{\text{new}} - x_i) = 0$$

$$\sum_{x_i \in S_j} \frac{k(x_i, c_j^{\text{old}})}{\sqrt{\left(\kappa - k(x_i, c_j^{\text{old}})\right)}} c_j^{\text{new}} = \sum_{x_i \in S_j} \frac{k(x_i, c_j^{\text{old}})}{\sqrt{\left(\kappa - k(x_i, c_j^{\text{old}})\right)}} x_i$$

$$c_j^{\text{new}} = \frac{\sum_{x_i \in S_j} w_i x_i}{\sum_{x_i \in S_j} w_i},$$

where we have defined weights

$$w_i := \frac{k(x_i, c_j^{\text{old}})}{\sqrt{\left(\kappa - k(x_i, c_j^{\text{old}})\right)}}.$$

Bibliography

[1] Alpcan, T., "Dual control with active learning using Gaussian process regression," Arxiv 1105.2211v1, Technical University Berlin, Berlin, 2011.

[2] Aronszajn, N., "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, pp. 337–404, may 1950.

[3] Aström, K. J. and Wittenmark, B., *Adaptive Control*. Readings: Addison-Weseley, 2nd ed., 1995.

[4] Baker, C., *The Numerical Treatment of Integral Equations*. Oxford, Clarendon Press, 1977.

[5] Belkin, M. and Niyogi, P., "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[6] Bernard, C. and Slotine, J., "Adaptive control with multiresolution bases," in *Proceedings of the 36th IEEE Conference on Decision and Control, 1997*, vol. 4, pp. 3884 –3889, Dec 1997.

[7] Bishop, C., *Pattern recognition and machine learning*. Springer New York, 2006.

[8] Boyd, S. and Sastry, S., "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, pp. 629–639, 1986.

[9] Brand., M., "Fast online svd revisions for lightweight recommender systems," in *Proc. of SIAM Int. Conf. on Data Mining*, 2003.

[10] Brand, M., "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, pp. 20–30, May 2006.

[11] Burges, C., "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[12] Calise, A., Hovakimyan, N., and Idan, M., "Adaptive output feedback control of nonlinear systems using neural networks," *Automatica*, vol. 37, no. 8, pp. 1201–1211, 2001. Special Issue on Neural Networks for Feedback Control.

[13] Candela, J. and Rasmussen, C., "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[14] Cannon, M. and Slotine, J., "Space-frequency localized basis function networks for nonlinear system estimation and control," *Neurocomputing*, vol. 9, no. 3, pp. 293 – 342, 1995. Control and Robotics, Part III.

[15] Chen, Y., Welling, M., and Smola, A., "Super-samples from kernel herding," in *Uncertainty in Artificial Intelligence*, 2010.

[16] Chen, Y. H., "Adaptive robust control of uncertain systems with measurement noise," *Automatica*, vol. 28, no. 4, pp. 715–728, 1992.

[17] Chowdhary, G., *Concurrent Learning for Convergence in Adaptive Control Without Persistency of Excitation*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2010.

[18] Chowdhary, G. and Johnson, E. N., "Concurrent learning for convergence in adaptive control without persistency of excitation," in *IEEE Conference on Decision and Control*, pp. 3674–3679, 2010.

[19] Chowdhary, G., How, J., and Kingravi, H., "Model reference adaptive control using nonparametric adaptive elements," in *Conference on Guidance Navigation and Control*, (Minneapolis, MN), AIAA, August 2012. Invited.

[20] Chowdhary, G. and Johnson, E. N., "A singular value maximizing data recording algorithm for concurrent learning," in *American Control Conference*, 2011. submitted.

[21] Chung, F., *Spectral Graph Theory*. Regional Conference Series in Mathematics, 1997.

[22] Coifman, R. and Lafon, S., "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 5–30, 2006.

[23] Cortes, C. and Vapnik, V., "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[24] Csató, L. and Opper, M., "Sparse on-line Gaussian processes," *Neural Computation*, vol. 14, no. 3, pp. 641–668, 2002.

[25] Deisenroth, M. P., *Efficient Reinforcement Learning using Gaussian Processes*. Karlsruhe, Germany: KIT Scientific Publishing, 1 ed., 2010.

[26] Donoho, D. L. and Grimes, C., "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003.

[27] Drineas, P. and Mahoney, M., "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *Journal of Machine Learning Research*, vol. 6, no. 10, pp. 2153–2175, 2005.

[28] Dudley, R., "The sizes of compact subsets of hilbert space and continuity of gaussian processes," *Journal of Functional Analysis*, vol. 1, no. 3, pp. 290–330, 1967.

[29] Durham, W., "Constrained control allocation," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 16, pp. 717–772, 1993.

[30] Folland, G., *Real analysis: modern techniques and their applications*, vol. 2. Wiley New York, 1984.

[31] Freedman, D. and Kisilev, P., "KDE Paring and a faster mean shift algorithm," *SIAM Journal of Imaging Sciences*, vol. 3, no. 4, pp. 878–903, 2010.

[32] Gelman, A., Carlin, J., Stern, H., and Rubin, D., *Bayesian Data Analysis.* Chapman and Hall, 2nd ed., 2004.

[33] Gredilla, M., Candela, J., Rasmussen, C., and Figueiras-Vidal, A., "Sparse spectrum Gaussian process regression," *Journal of Machine Learning Research*, vol. 11, pp. 1865–1881, 2010.

[34] Gretton, A., Bousquet, O., Smola, A. J., and Schölkopf, B., "Measuring statistical dependence with Hilbert-Schmidt norms," in *Algorithmic Learning Theory*, pp. 63–77, 2005.

[35] Ham, J., Lee, D., Mika, S., and Schölkopf, B., "A kernel view of the dimensionality reduction of manifolds," in *Proceedings of the International Conference on Machine Learning*, 2004.

[36] Hsu, D., Kakade, S. M., and Zhang, T., "A spectral algorithm for learning hidden markov models," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1460–1480, 2012.

[37] Ioannou, P. A. and Sun, J., *Robust Adaptive Control*. Upper Saddle River: Prentice-Hall, 1996.

[38] Ioannou, P. A. and Sun, J., *Robust adaptive control*. Dover, 2012.

[39] Johnson, E. and Kannan, S., "Adaptive trajectory control for autonomous helicopters," *Journal of Guidance Control and Dynamics*, vol. 28, pp. 524–538, May 2005.

[40] Johnson, E. N., *Limited Authority Adaptive Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2000.

[41] Kannan, S., *Adaptive Control of Systems in Cascade with Saturation*. PhD thesis, Georgia Institute of Technology, Atlanta Ga, 2005.

[42] Khalil, H. K., *Nonlinear systems*, vol. 3. Prentice hall Upper Saddle River, 2002.

[43] Kharisov, E. and Hovakimyan, N., "Application of l1 adaptive controller to wing-rock," in *AIAA Infotech@Aerospace*, april 2010.

[44] Khasminksii, R., *Stochastic Stability of Differential Equations*. Berlin, Germany: Springer, 2 ed., 2012.

[45] Kim, N., *Improved Methods in Neural Network Based Adaptive Output Feedback Control, with Applications to Flight Control*. PhD thesis, Georgia Institute of Technology, Atlanta GA, 2003.

[46] Kim, Y. H. and Lewis, F., *High-Level Feedback Control with Neural Networks*, vol. 21 of *Robotics and Intelligent Systems*. Singapore: World Scientific, 1998.

[47] Kingravi, H. A., Chowdhary, G., Vela, P. A., and Johnson, E. N., "Reproducing kernel Hilbert space approach for the online update of radial bases in neuro-adaptive control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1130 –1141, July 2012.

[48] Ko, J., Klein, D., Fox, D., and Haehnel, D., "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp," in *IEEE International Conference on Robotics and Automation*, pp. 742–747, IEEE, 2007.

[49] Ko, J. and Fox, D., "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, pp. 75–90, 2009.

[50] Koltchinskii, V., *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: École dÉté de Probabilités de Saint-Flour XXXVIII-2008*, vol. 2033. Springer, 2011.

[51] Kreisselmeier, G. and Narendra, K. S., "Stable model reference adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. AC-27, no. 6, pp. 1169–1175, 1982.

[52] Kushner, H. J., *Stochastic Stability and Control*. New York, NY: Academic Press, 1967.

[53] Khn, T., "Covering numbers of gaussian reproducing kernel hilbert spaces," *Journal of Complexity*, vol. 27, no. 5, pp. 489 – 499, 2011.

[54] Lafon, S. S., *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, 2004.

[55] Lee, J. and Verleysen, M., *Nonlinear Dimensionality Reduction*. Springer, 2007.

[56] Lewis, F. L., "Nonlinear network structures for feedback control," *Asian Journal of Control*, vol. 1, pp. 205–228, 1999. Special Issue on Neural Networks for Feedback Control.

[57] Liu, W., Principe, J. C., and Haykin, S., *Kernel Adaptive Filtering: A Comprehensive Introduction*. Hoboken, New Jersey: Wiley, 2010.

[58] Mercer, J., "Functions of positive and negative type and their connection with the theory of integral equations," *Philosophical Transactions of the Royal Society A*, vol. 209, pp. 441–458, 1909.

[59] Micchelli, C. A., "Interpolation of scattered data: distance matrices and conditionally positive definite functions," *Constructive Approximation*, vol. 2, no. 1, pp. 11–22, 1986.

[60] Miyahara, Y., "Ultimate boundedness of the systems governed by stochastic differential equations," *Nagoya Math Journal*, vol. 47, pp. 111–144, 1972.

[61] Monahemi, M. M. and Krstic, M., "Control of wingrock motion using adaptive feedback linearization," *Journal of Guidance Control and Dynamics*, vol. 19, no. 4, pp. 905–912, 1996.

[62] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, S., and Schölkopf, B., "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–202, 2001.

[63] Murray-Smith, R. and Sbarbaro, D., "Nonlinear adaptive control using non-parametric Gaussian process prior models," in *15th Triennial World Congress of the International Federation of Automatic Control*, International Federation of Automatic Control (IFAC), 2002.

[64] Murray-Smith, R., Sbarbaro, D., Rasmussen, C., and Girard, A., "Adaptive, cautious, predictive control with Gaussian process priors," in *13th IFAC Symposium on System Identification* (de Hof, P. V., Wahlberg, B., and Weiland, S., eds.), pp. 1195–1200, Elsevier Science, 2003.

[65] Nadler, B., Lafon, S., Coifman, R., and Kevrekidis, I., "Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 113–127, 2006.

[66] Nardi, F., *Neural Network based Adaptive Algorithms for Nonlinear Control*. PhD thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA, 2000.

[67] Narendra, K. and Annaswamy, A., "A new adaptive law for robust adaptation without persistent excitation," *IEEE Transactions on Automatic Control*, vol. 32, no. 2, pp. 134–145, 1987.

[68] Narendra, K. S. and Annaswamy, A. M., "Robust adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. AC-31, no. 4, pp. 306–315, 1986.

[69] Narendra, K. S. and Annaswamy, A. M., *Stable Adaptive Systems*. Englewood Cliffs: Prentice-Hall, 1989.

[70] Narendra, K., "Neural networks for control theory and practice," *Proceedings of the IEEE*, vol. 84, pp. 1385 –1406, oct 1996.

[71] Nguyen-Tuong, D., Schölkopf, B., and Peters, J., "Sparse online model learning for robot control with support vector regression," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 3121–3126, 2009.

[72] Nguyen-Tuong, D. and Peters, J., "Using model knowledge for learning inverse dynamics," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2677 –2682, may 2010.

[73] Park, J. and Sandberg, I., "Universal approximation using radial-basis-function networks," *Neural Computation*, vol. 3, pp. 246–257, 1991.

[74] Patiño, H., Carelli, R., and Kuchen, B., "Neural networks for advanced control of robot manipulators," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 343–354, 2002.

[75] Patino, D. and Liu., D., "Neural network based model reference adaptive control system," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 30, no. 1, pp. 198–204, 2000.

[76] Quinlan, J., "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[77] Rasmussen, C. and Kuss, M., "Gaussian processes in reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 16, pp. 751–759, 2004.

[78] Rasmussen, C. and Williams, C., *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[79] Reinhardt, H.-J., *Analysis of Approximation Methods for Differential and Integral Equations*. Springer-Verlag, 1985.

[80] Rosasco, L., Belkin, M., and Vito, E., "On learning with integral operators," *Joural of Machine Learning Research*, vol. 11, pp. 905–934, 2010.

[81] Rosenblatt, F., "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological Review*, vol. 65, no. 6, p. 386, 1958.

[82] Roweis, S. T. and Saul, L. K., "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[83] Rumelhart, D., Hinton, G., and Williams, R., "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[84] Saad, A. A., *Simulation and analysis of wing rock physics for a generic fighter model with three degrees of freedom*. PhD thesis, Air Force Institute of Technology, Air University, Wright-Patterson Air Force Base, Dayton, Ohio, 2000.

[85] Sanner, R. and Slotine, J.-J., "Gaussian networks for direct adaptive control," *IEEE Transactions on Neural Networks*, vol. 3, pp. 837 –863, nov 1992.

[86] Sarwar, B., G.Karypis, Konstan, J., and Reidl, J., "Incremental singular value decomposition algorithms for highly scalable recommender systems.," in *Int. Conf. on Computer and Information Technology*, 2002.

[87] Saunders, C., Gammerman, A., and Vovk, V., "Ridge regression learning algorithm in dual variables," in *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*, pp. 515–521, Morgan Kaufmann, 1998.

[88] Scholkopf, B., Herbrich, R., and Smola, A., "A generalized representer theorem," in *Computational Learning Theory* (Helmbold, D. and Williamson, B., eds.), vol. 2111 of *Lecture Notes in Computer Science*, pp. 416–426, Springer Berlin / Heidelberg, 2001.

[89] Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K., Rätsch, G., and Smola, A., "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[90] Scholköpf, B. and Smola, A., *Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT press, 2002.

[91] Scholköpf, B., Smola, A., and Müller, R., "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[92] Shawe-Taylor, J. and Cristianini, N., *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press, 2004.

[93] Singh, S. N., Yim, W., and Wells, W. R., "Direct adaptive control of wing rock motion of slender delta wings," *Journal of Guidance Control and Dynamics*, vol. 18, no. 1, pp. 25–30, 1995.

[94] Smale, S. and Zhou, D.-X., "Geometry on probability spaces," *Constructive Approximation*, vol. 30, no. 3, pp. 311–323, 2009.

[95] Smola, A., Gretton, A., Song, L., and Scholkopf, B., "A Hilbert space embedding for distributions," *Algorithmic Learning Theory*, 2007.

[96] Song, L., *Learning via Hilbert space embedding of distributions*. PhD thesis, University of Sydney, Sydney, Australia, 2008.

[97] Song, L., Boots, B., Siddiqi, S. M., Gordon, G. J., and Smola, A. J., "Hilbert space embeddings of hidden markov models," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 991–998, 2010.

[98] Song, L., Fukumizu, K., and Gretton, A., "Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models," *Signal Processing Magazine, IEEE*, vol. 30, no. 4, pp. 98–111, 2013.

[99] Song, L., Huang, J., Smola, A., and Fukumizu, K., "Hilbert space embeddings of conditional distributions with applications to dynamical systems," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 961–968, ACM, 2009.

[100] Steinwart, I. and Christmann, A., *Support vector machines*. Springer, 2008.

[101] Sundararajan, N., Saratchandran, P., and Yan, L., *Fully Tuned Radial Basis Function Neural Networks for Flight Control*. Springer, 2002.

[102] Suykens, J., Vandewalle, J., and Moor, B. D., *Artificial Neural Networks for Modeling and Control of Non-Linear Systems*. Norwell: Kluwer, 1996.

[103] Talwalkar, A., *Matrix Approximation for Large-scale Learning*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, New York, NY, 2010.

[104] Tao, G., *Adaptive Control Design and Analysis*. New York: Wiley, 2003.

[105] Thorstensen, N., Segonne, F., and Keriven, R., "Pre-image as karcher mean using diffusion maps: Application to shape and image denoising," *Scale Space and Variational Methods in Computer Vision*, pp. 721–732, 2009.

[106] Vapnik, V., *The nature of statistical learning theory*. Springer, 1999.

[107] Vito, E. D., Rosasco, L., Caponnetto, A., Giovannini, U. D., and Odone, F., "Learning from examples as an inverse problem," *Journal of Machine Learning Research*, vol. 6, pp. 883–904, Dec. 2005.

[108] Volyanskyy, K., Haddad, W., and Calise, A., "A new neuroadaptive control architecture for nonlinear uncertain dynamical systems: Beyond σ and e-modifications," *IEEE Transactions on Neural Networks*, vol. 20, no. 11, pp. 1707–1723, 2009.

[109] Wang, X., Tino, P., Fardal, M., Raychaudhury, S., and Babul, A., "Fast parzen window density estimator," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 3267–3274, 2009.

[110] Wasserman, L., *All of Statistics*. Springer, 2004.

[111] Wendland, H., *Scattered data approximation*, vol. 17. Cambridge University Press Cambridge, 2005.

[112] Williams, C. and Seeger, M., "The effect of the input density distribution on kernel-based classifiers," in *Proceedings of the International Conference on Machine Learning*, pp. 1159–1166, 2000.

[113] Williams, C. and Seeger, M., "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*, pp. 682–688, 2001.

[114] Yucelen, T. and Calise, A., "A derivative-free model reference adaptive controller for the generic transport model," in *AIAA Guidance, Control and Navigation Conference*, (Toronto, Canada), August 2010. invited.

[115] Zhang, K. and Kwok, J., "Improved Nyström low rank approximation and error analysis," in *Proceedings of the International Conference on Machine Learning*, pp. 1232–1239, 2008.

[116] Zhang, K. and Kwok, J., "Density-weighted Nyström method for computing large kernel eigensystems," *Neural Computation*, vol. 21, no. 1, pp. 1299–1319, 2010.

[117] Zhang, M.-G. and Liu, W.-H., "Single neuron PID model reference adaptive control based on RBF neural networks.," in *2006 International Conference on Machine Learning and Cybernetics*, pp. 3021–3025, 2006.

[118] Zhang, T., Ge, S., and Hang, C., "Direct adaptive control of non-affine nonlinear systems using multilayer neural networks," in *American Control Conference, 1998. Proceedings of the 1998*, vol. 1, pp. 515 –519 vol.1, jun 1998.

[119] Zwald, L. and Blanchard, G., "On the convergence of eigenspaces in kernel principal component analysis," in *Advances in Neural Information Processing Systems*, 2005.

[120] Zwald, L., Bousquet, O., and Blanchard, G., "Statistical properties of kernel principal component analysis," in *Learning theory*, pp. 594–608, Springer, 2004.

Reduced-Set Models for Improving the Training and Execution Speed of Kernel Methods

Hassan A. Kingravi

179 Pages

Directed by Professor Patricio A. Vela

This thesis aims to contribute to the area of kernel methods, which are a class of machine learning methods known for their wide applicability and state-of-the-art performance, but which suffer from high training and evaluation complexity.

The work in this thesis utilizes the notion of *reduced-set models* to alleviate the training and testing complexities of these methods in a unified manner. In the first part of the thesis, we use recent results in kernel smoothing and integral-operator learning to design a generic strategy to speed up various kernel methods. In Chapter 3, we present a method to speed up kernel PCA (KPCA), which is one of the fundamental kernel methods for manifold learning, by using reduced-set density estimates (RSDE) of the data. The proposed method induces an integral operator that is an approximation of the ideal integral operator associated to KPCA. It is shown that the error between the ideal and approximate integral operators is related to the error between the ideal and approximate kernel density estimates of the data. In Chapter 4, we derive similar approximation algorithms for Gaussian process regression, diffusion maps, and kernel embeddings of conditional distributions.

In the second part of the thesis, we use reduced-set models for kernel methods to tackle online learning in model-reference adaptive control (MRAC). In Chapter 5, we relate the properties of the feature spaces induced by Mercer kernels to make a connection between persistency-of-excitation and the budgeted placement of kernels to minimize tracking and modeling error. In Chapter 6, we use a Gaussian process (GP) formulation of the modeling error to accommodate a larger class of errors, and design a reduced-set algorithm to learn a GP model of the modeling error. Proofs of

stability for all the algorithms are presented, and simulation results on a challenging control problem validate the methods.